FINAL REPORT

VOLUME 3

TASK 3: PARALLEL FUNCTION PROCESSOR TECHNOLOGY

CLIN 0006

November 2, 1990

MACROSTRUCTURE LOGIC ARRAYS

Contract No. DASG60-85-C-0041

Sponsored By

The United States Army Strategic Defense Command

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332 - 0540

Contract Data Requirements List Item F006

Period Covered: 1985-1990

Type Report: Final

AD-A228 564

90 11- 9 · 019

DISCLAIMER

DISCLAIMER STATEMENT - The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

DISTRIBUTION CONTROL

(1) **DISTRIBUTION STATEMENT** - Approved for public release; distribution is unlimited.

(2) This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227 - 7013, October 1988.

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT 1)Approved for public release; dis- tribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	2) (continued on reverse side)
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION School of Electrical Eng. Georgia Tech	6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION U.S. Army Strategic Defense Command
6c ADDRESS (City, State, and ZIP Code) Atlanta, Georgia 30332		7b ADDRESS (City, State, and ZIP Code) P.O. Box 1500 Huntsville, AL 35807-3801

8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DASG60-85-C-0041

8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

Macrostructure Logic Arrays Volume 3, Task 3

12 PERSONAL AUTHOR(S)
C. O. Alford

13a TYPE OF REPORT Final	13b TIME COVERED FROM 6/28/85 TO 11/2/90	14 DATE OF REPORT (Year, Month, Day) November 2, 1990	15 PAGE COUNT 41

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Task 3: Parallel Function Processor Technology
1. Introduction
2. PFP
  2.1 Hosts
  2.2 Processors
  2.3 Crossbar
  2.4 Interfaces
  2.5 System Documentation
  2.6 PFP Training
  2.7 PFP Testing
  2.8 Existing Systems

3. Performance Benchmarks
  3.1 Spinning Missile Simulation
  3.2 Target Initialization
     Benchmark
  3.3 Satellite Altitude Control
  3.4 High-Endoatmospheric
     Simulation
  3.5 Exoatmospheric Simulation
4. References

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL

DD Form 1473, JUN 86   Previous editions are obsolete

istribution statement continued
) This material may be reproduced by or for the U.S. Government pursuant to
   the copyright license under the clause at DFARS 252.227-7013, October 1988.

FINAL REPORT
VOLUME 3
TASK 3: PARALLEL FUNCTION PROCESSOR TECHNOLOGY

CLIN 006

November 7, 1990

Authors
Michael B. Woods, Richard M. Pitts, Philip R. Bingham, David Palusky

.

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332 - 0540

Eugene L. Sanders   Cecil O. Alford

USASDC   Georgia Tech

Contract Monitor   Project Director

Table of Contents

1. Introduction

The DETL (Digital Emulation Technology Laboratory) simulation hardware centers on the development, implementation, and use of the Parallel Function Processor (PFP). The PFP is a 64 processor digital computer for use in computationally intensive applications that can be partitioned into functional blocks. The processors are grouped in two 32 processor clusters running from one common host. Each 32 processor cluster is connected by a crossbar switch. All inter-processor communication takes place over the crossbar(s). Simultaneous transfers may take place independently and switch patterns may be changed every cycle. In order to program the machine correctly, all inter-processor communication and data transfer lengths must be known beforehand.

The PFP has been designed to accomodate "hardware in the loop" simulations running in real time. Actual hardware components may first be simulated on one or more processors and later replaced with actual hardware interfaced to specified crossbar ports. The inputs and outputs to/from the device will appear identical to those it would see in an actual system.

Figure 1.1 illustrates the basic PFP architectural concept. Figure 1.2 illustrates a front view of the actual machine. A deeper level of architectural detail can be found in the final report for FY89 [1], contract number DASG60-85-C-0041.

1.1 History

The PFP project is the outgrowth of doctoral research work that was initiated by M. R. McQuade [2] and continued by J. O. Hamblen [3], both under the direction of Dr. C. O. Alford at Georgia Tech. At the initial awarding of this contract in 1985, a 32 processor unit based on 5 Mhz 8086 based processors had been prototyped and was hosted by an Intel Series III computer. Since that time, all portions of the system have been re-designed to provide increased performance, increased reliability, increased modularity, decreased size, easier programmability, and easier interfacing to external equipment. Multiple systems have now been built. Each system reflects the latest upgrades up to the time of its construction.. The highest performance system developed is capable of supporting 64 high speed processors for an estimated system performance of 480 MFLOPS. Through each upgrade, the systems have adhered to the same basic architectural concepts.

Spin offs from this initial project have included Georgia Tech's Guidance, Navigation, and Control chip (GN&C) set, a signal processing chip set, and the Seeker/Scene Emulator (SSE). Each of these pieces of hardware is architecturally unique to fit its function, and each has been or is being designed to work in conjunction with the PFP and with each other.

1.2. Objectives

Within DETL, there are currently two main hardware systems: The PFP and the SSE. (The SSE is covered in Volume 2 of this report.) The two systems are designed to function together as a simulation/emulation facility for kinetic energy weapons systems. The principal objectives of the DETL are as follows:

- Provide facilities for 6-DOF KEW emulation

- Provide real-time capability in excess of 2000 Hz

1
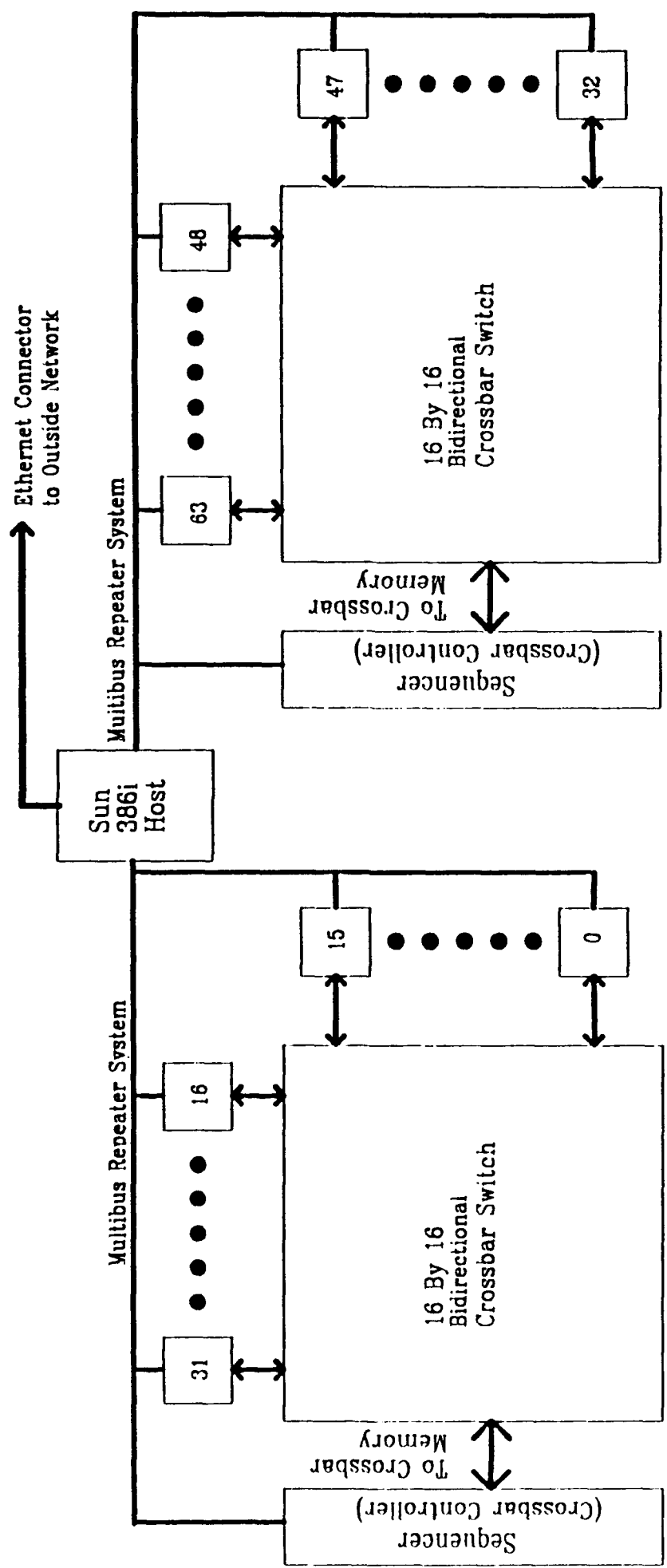
Figure 1.1 Basic PFP Architecture
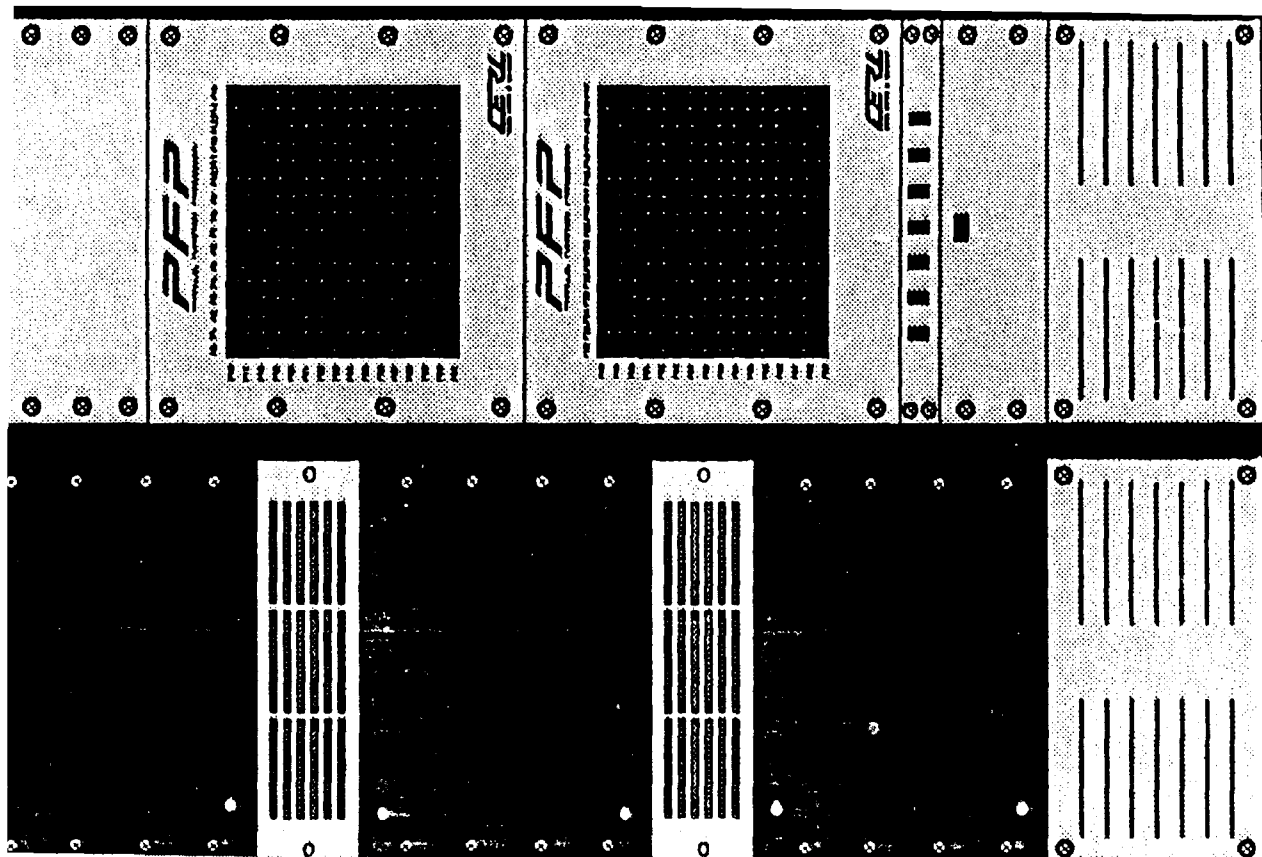
- Provide support for nonlinear functions

- Provide real-time emulation of IR FPA seekers

- Provide a facility for testing and verification of GN&C processors

Real-time emulation of IR FPA seekers is primarily the responsibility of the Seeker/Scene Emulator. The other objectives are primarily the responsibility of the PFP.

1.3. Requirements

The requirements for this portion of the contract have been to develop the PFP and SSE hardware and software to meet the objectives set forth in Section 1.2. This has included custom hardware and software development, and the integration of commercially available hardware and software products into our environment. Custom hardware developed at the board level includes processors, networks and netwotk control, interfaces to external equipment, and the interconnecting circuitry needed to integrate these components into a complete system. In addition the system hardware has been designed and documented to meet the following requirements:

- Development and delivery of a complete set of documentation describing the requirements for manufacture and acceptance testing of the PFP so that the units can be reproduced.

- Development and delivery of a hardware operator's manual for the PFP.

- Development and delivery of a software programmer's manual for the PFP.

-Development of a set of system diagnostics to aid in testing newly constructed systems and for use in debugging.

- Development and execution of a PFP training session.

- Development and execution of PFP reliability and thermal testing.

- Development and implementation of a materials management system for ordering PFP parts at the system, subsystem, and component levels.

- In house manufacture of a prototype PFP unit for delivery to USADC's KDEC facility in Huntsville, Al.

2. PFP

2.1 Hosts

The PFP host is the platform where programs are written, stored, and compiled for the PFP and loaded to the PFP. It is connected to the PFP through a custom designed Multibus repeater link. There are currently two PFP hosts being used, the Intel 310 and the Sun 386i. The Intel 310 (the original replacement for the Intel Series III) is currently the primary one in use, but is in the process of being phased out. The Sun 386i is meant as its eventual replacement, and is in the final stages of development.

2.1.1 Intel 310

The Intel 310 is a commercially available 80286 based computer available from Intel Corporation. It is interfaced to the PFP through a custom designed board that resides inside the 310 and interfaces to each Multibus chassis in the PFP The 310 uses Intel's iRMX II.4 real time multitasking operating system. Custom libraries containing all the necessary routines for communications with the PFP and communication over the crossbar have been developed, and can be used in conjunction with a variety of high level languages. In particular, Pascal, FORTRAN, C, and PL/M languages are available for use in conjunction with 80286 and 80386 based processors. The Intel 310 also contains limited software support for the custom built GT-FPP/3 floating point processor. This consists of a Georgia Tech compiler that implements a subset of Pascal and the necessary I/O functions. Use of the Intel 310 system software in conjunction with the PFP is documented in the PFP Programmers Manual [4]. Copies of the actual software are included in the FY89 final report [1] Volume 4, Appendix C.

2.1.2 Sun 386i

The Sun 386i has been developed as the replacement for the Intel 310 and uses a Unix operating system. The major development for the system has been in developing the utilities necessary to interface to the PFP and in developing full compilers for the custom GT-FPP/3 processor. In addition, support for the 80386 family of processors has been retained. Both the Ada and C languages are to be supported. A custom C compiler has been written for the board and is in the final stages of debug. Ada will be supported through the use of a commercially available Ada to C translator.

Three Sun 386i systems are currently being used in conjunction with PFPs. One is being used for the compiler development. Two are being used for application software development, particularly for putting portions of the EXOSIM missile simulation on the GT-FPP/3 processors.

2.2 Processors

There are currently 3 main processors being used in the PFP systems. They are the Intel iSBC286/12, the Intel iSBC386/12, and the Georgia Tech GT-FPP/3. An earlier 8086 based processor (iSBC86/12) was used in the original version of the system but has been phased out. In order to support the commercially available iSBC286/12 and iSBC386/12 processors, modifications were made to the on board firmware. Any other commercially available Multibus I processor is a potential candidate for PFP support provided that similar modifications can be made. For example, as soon as a Multibus I 80486 based board comes on the market, it can be integrated into the PFP environment by making similar firmware changes.

2.2.1 iSBC86/12

The iSBC86/12 processor was the first processor used in the PFP system. The board was based on a 5 Mhz 8086 processor with an 8087 math co-processor. The board served as the main system processor until 1987 when the iSBC286/12 was integrated. Details on the use of the iSBC86/12 can be found in the FY87 final report [4], Vol. 1, p. 30.

2.2.2 iSBC286/12

The Intel iSBC286/12 is a commercially available processor board based on an 8 Mhz 80286 processor with an 80287 math co-processor and 1 megabyte of memory. Modifications have been made to the on

5

board firmware in order to use it in the PFP environment. Details on the modifications can be found in the PFP Technical Data Package [6], Volume 1 [4] and The FY89 final report [1], Volume 4, Appendix A. Programming languages available for the board include Pascal, FORTRAN, C, and PL/M. All of the compilers are commercially available from Intel.

The iSBC286/12 is interfaced to the crossbar through the use of a custom designed piggyback board that mates to the Intel iSBX port connector called the GT-XI286/1. The iSBX port is an Intel standard I/O port found on many Multibus I processor boards. Details on the GT-XI286/1 can be found in the Technical Data Package [6] Volume 1.

The iSBC286/12 was the main processor used in the DETL for several years. Recently, the 80386 based boards and the GT-FPP/3 have taken over the main computing load. These processors are still used for simulation code development, with the resulting code taken and recompiled for one of the faster processors.

2.2.3 iSBC386/12

The iSBC386/12 is a commercially available processor board based on a 20 Mhz 80386 with an 80387 math co-processor and 1 megabyte of memory. Modifications similar to those for the iSBC286/12 were made in order to use it in the PFP environment. The same languages available for the iSBC286/12 are also available for the iSBC386/12.

The iSBC386/12 is interfaced to the crossbar using an iSBX piggyback board that is functionally equivalent to the one used on the iSBC286/12. The board layout was re-arranged because it interfered with other components on the board.

There are presently 32 iSBC386/12 processors being used in the DETL. Support from the Intel 310 hosted environment is complete and in every day use. Support from the Sun386i hosted environment is functional, but not fully debugged.

2.2.4 GT-FPP/3

The GT-FPP/3 is a custom built 32 bit processor capable of sustained 8 megaflop performance. The processor is based on the AMD29325 arithmetic logic unit. The board utilizes two separate memory units, the instruction memory and the data memory. Instruction fetches are reduced to one memory access by utilizing wide and parallel instructions to control the various processing elements of the floating point processor. The data memory is designed with three ports, so that two operands may be fetched and one result stored simultaneously. Furthermore, pipelining is used to overlap the instruction and operand fetches. Further details c the GT-FPP/3 can be found in the PFP Technical Data Package [6], Volume 1.

Several add on boards are available for the GT-FPP/3. The GT-DT2/1 is an add on memory board that supplies all of the data memory needed by the processor. The board consists of two read ports and one write port. The read ports give simultaneous access to two 32 bit operands while the write port stores a 32 bit result. The board is designed around 8 IDT7132 dual port 2k by 8 memory chips. Details are available in the Technical Data Package, Volume 1.

The GT-FFS/1 provides hardware assisted calculation of the functions sine, cosine, tangent, arcsine, arccosine, arctangent, exponential, natural logarithm, inverse, and square root, by storing pre-computed values in permanent tables. The tables are stored in ROM for each function. In order to calculate a given

function for the variable x, x is used as a pointer into the table to retrieve a function value. The FFS/1 is also included in the Technical Data Package [6], Volume 1. It is added through a dedicated connector on the main board.

The GT-FFL/1 is an add on board that works similarly to the GT-FFS/1 (it interfaces to the same connector on the GT-FPP/3) but stores table values in RAM instead of ROM. This way tables may be loaded for any function. Each GT-FFL/1 is capable of supporting two functions, and can handle the storage for 64k 32bit values per function.

The GT-FPX/1 provides double precision calculations. It is inserted in the socket originally for the 29325. Work is currently under way modify the Georgia Tech C compiler to support double precision calculations.

The GT-FPP/3 software has primarily been developed in the Sun hosted environment, with the C compiler starting to come into every day use. The Ada translator work is still under way. The Pascal subset that was implemented under the Intel 310 environment was used for several benchmarks, inluding the spinning missile benchmark which will be discussed later in this report.

2.3 Crossbar

The crossbar and sequencer combination make up the processor interconnection network used in the PFP. The crossbar contains the actual data paths and all switch points between processors. The sequencer is the device that sets the switch patterns and regulates the communication between processors. The crossbar/sequencer pair is programmed using a custom crossbar compiler. Details on how to use the crossbar compiler can be found in the PFP Software Programmer's Manual [4].

2.3.2 GT-SEQ/2 Sequencer

Communication between processors in the PFP is asynchronous. A cycle of simultaneous conversations only occurs when all involved processors are ready, and the actual communication uses a full handshake to insure reliable data transfer. The Sequencer is the board used to monitor the readiness of all processors and control program flow from one cycle to the next. The sequencer also requlates the handshaking and acts as a supervisor to the crossbar, providing pointers to crossbar memory, which stores the switchpoint configurations required for each cycle.

The basic design of the GT-SEQ/2 sequencer is much the same as the original version [5] prototyped under the direction of Dr. J. O. Hamblen. The most significant differences are 1) a 2:1 reduction in size, 2) differential drivers and receivers on all handshake lines for inreased noise immunity, 3) increased memory for more instructions, and 4) full conformability to the Multibus I specification for a 16 bit slave. Further details on the GT-SEQ/2 sequencer can be found in the PFP Technical Data Package [6], Volume 1. A detailed timing analysis of the GT-SEQ/2 can be found in the FY89 final report [1], Volume 1, pp. 29 - 37.

2.3.2 GT-XB/2 Crossbar Switch

The crossbar is a 16 by 16 switch matrix made up of 4 interconnected 8 by 8 switch matrices. Only one 8 by 8 switch board was designed - the 4 sections are identical except for the board address switch setting. Each 8 by 8 switch board (GT-XB/2) is matched with a piggyback board (GT-DXB/1) to allow the use of more connectors. The board pairs are connected through the use of a custom crossbar backplane to make

7

up the full 16 by 16 matrix. Each processor port is a bidirectional 16 bit parallel data path so that 32 processors can be connected to one crossbar unit. Each processor may send or receive data on a particular cycle, but not both. The crossbar contains a full broadcast capability so that any 31 processors can receive data from any one processor simultaneously.

The backplane was designed to support two full crossbars. Two full crossbars can be housed in one card cage which is approximately 37 cm by 40 cm by 49 cm. This is approximately a 16:1 size reduction over the original crossbar unit. Details on the GT-XB/2 and the GT-DXB/1 can be found in the PFP Technical Data Package [6], Volume 1.

2.4. Interfaces

The ability to interface to other computers and to external hardware is crucial in a testing environment. In a testing environment, the PFP may be used to simulate certain components of a system. When an actual component is ready to test, the code (and processor) simulating that component may be removed, and the actual component interfaced to the system in its place. Thus the component can be tested while the rest of the system remains an unmodified simulation.

2.4.1. GT-XB/2 Crossbar Switch

Interfacing actual components, such as a flight computer or IMU, directly to the crossbar is accomplished by matching the data lines of the device to the 16 bit parallel crossbar port and the handshake lines of the device to the 4 handshake lines of the sequencer. The complexity of the interface is partially dependent on the specific device, but any device with digital outputs should not present a problem. Devices with parallel output ports are particularly straightforward.

As an example, a Honeywell built SANDAC S5 flight computer has been interfaced directly to the crossbar. The actual hardware interface consists of 5 chips (2 registers, 2 tranceivers, and 1 EPLD) and the matching connectors. A complete 27 processor simulation was programmed on the system. The flight control program was first run on a GT-FPP/3 processor. The same program was then run on the SANDAC in place of the GT-FPP/3. The only system change involved re-compiling the crossbar code so that the data from the flight control computer came from the correct port. Details on the entire simulation can be found in [7] pp 938 - 943.

2.4.2. GT-ADDA/2 Analog I/O Board

The GT-ADDA/2 consists of four 12 bit analog input channels and four 12 bit analog output channels. The board occupies one crossbar port and fits in a standard Multibus I card cage. Although the board meets the Multibus I form factor, no Multibus interface exists on the board. The Multibus P1 connector is used only for power, ground, and clock connections. All communication with the PFP takes place through the crossbar port.

The digital to analog portion of the board consists of four 12 bit digital to analog converters. Each channel can be individually jumpered for one of three output ranges: -10 volts to 10 volts, 0 volts to 10 volts, or -5 volts to 5 volts. Conversion time is approximately 250 nanoseconds, with the actual delay between the crossbar write and write acknowledge signals being 100 nanoseconds. When coupled with a complete crossbar cycle, the actual time between consecutive writes is approximately 1.2 microseconds. The analog outputs are buffered with an output circuit based on an Analog Devices AD711 operational amplifier.

The analog to digital portion of the board consists of 4 input channels demultiplexed through one analog to digital converter. The 12 bit A/D converter is configured to convert analog values in the range of 0 to 10 volts. A programmable gain amplifier is present before the A/D converter so that smaller input ranges, for example 0 to +5 volts, may utilize the full range of the converter. For 0 to 5 volts the gain would be set to 2, for 0 to 2.5 volts the gain would be set to 4, etc. Actual conversion time on an A/D read is approximately 1 microsecond. When coupled with a complete crossbar cycle, the time between consecutive samples is approximately 2.5 microseconds.

Further details on the GT-ADDA/2 can be found in the PFP Technical Data Package [6], Vol. 1.

2.4.3 Array Interconnect

The GT-ARI/1 array interconnect allows for direct connection between multiple crossbars, thus allowing a processor on one crossbar to communicate directly with a processor on a second crossbar. Details on the GT-ARI/1 are avaulable in the PFP Technical Data Package [6], Vol. 1 and The FY89 final report [1] pp. 23 - 25.

2.4.4 PFP/SSE Interconnect

The GT-XIT/1 interface was developed to connect the Seeker/Scene Emulator directly to a crossbar port on the PFP. Details on this interface are available in the FY89 final report [1], p. 45.

2.4.5 Intel iSBX Port

The Intel iSBX port is a standard16 bit I/O port found on many Multibus I and Multibus II processors. Sections 2.2.1 and 2.2.2 explained how it was being used to interface the iSBC286/12 and iSBC386/12 processors to the crossbar. The same interface boards can be used on a variety of other processor boards so that they are immediate candidates for integrating into the PFP environment. Often a processor may have more than one iSBX port. A second iSBX port is a potential port for entering data directly into a PFP processor from an external source. This alternative is being looked at in conjunction with interfacing issues at the Arnold Engineering Development Center and the possible installation of a PFP unit there.

2.4.6 Ethernet

An ethernet connection is available on both the Intel 310 and the Sun386i. This allows files to be transferred between the PFP and other systems, and allows for remote log in capability. However, the two hosts do run different network protocols. The Intel 310 uses Intel's Openet protocol. In order to communicate with the PFP with the Intel 310 as host, the other machines on the network must also use it. Likewise, the Sun386i currently uses TCP/IP network protocol. In order for the other machines on the network to communicate with the PFP when the Sun is the host they must also use the TCP/IP protocol. Other network packages are available for the Sun including DECnet (which is commonly used with VAX systems) but have not been tried as of yet.

2.5. System Documentation

2.5.1. Technical Data Package

A four volume Technical Data Package [6] has been developed and delivered to USADC. The package describes the requirements for manufacture and acceptance of the PFP. Volume 1 is titled "System

9

Documeration". It contains all text on PFP assembly and each sub-assembly. Each sub-assembly is organized as a separate "User's Guide", including theory of operation, hardware options, assembly instructions, and programmable device listings.

Volume 2 is titled "Assembly Drawings". It contains all system level drawings and parts lists including all AC and DC chassis wiring, mechanical fabrication drawings, cable construction diagrams, subsystem placement, and all miscellaneous drawings.

Volume 3 is titled "Schematics". It contains all electrical schematics, assembly drawings, and parts lists for the circuit boards in the system. Each board is considered a sub-assembly.

Volume 4 is titled "Test Programs". It contains printouts of all PFP system and subssystem diagnostic and acceptance tests.

2.5.2. PFP Hardware Operation Manual

A PFP Hardware Operation Manual [8] has been written and delivered to USADC. The purpose of the manual is to give the PFP operator a functional understanding of how the PFP works, its capabilities, and how to use it. In addition, the manual explains how to run system diagnostics and how to locate errors based on the results. The PFP also contains two displays to aid in program debug and troubleshooting, 1) the crossbar status displays, and 2) the sequencer/processor transition boards. The manual goes through three examples showing how to read the displays and how to track programming bugs to a processing element based on what the displays read.

The first version of the manual has been delivered to USADC as a special technical report. The manual reflects the current system configuration. Since the PFP systems are continually being improved, minor changes to the manual are inevitable. For example, the current manual descibes an Intel 310 computer as the host. The Sun 386i host is fully functional from a hardware standpoint, but the system software support for it is not yet complete, thus Intel 310 is still the main host in use. A final version of the manual is due in 1991 which will reflect the current configuration as of that date, as well as changes deemed necessary from feedback on the original manual.

2.5.3. PFP Programmer's Manual

A Programmer's Manual [4] for the PFP has been written and delivered to USADC. The manual provides the information needed for a programmer to understand and program the Parallel Function Processor. Information on languages, syntax, and memory limits are presented. Additional information on how to use existing system software is also discussed.

The first version of this manual has been delivered as a special technical report. As with the hardware operation manual, a final version is due in 1991. The current version assumes the Intel 310 as host and the Intel family of processors as the processing elements. The final version should use the Sun 386i as host and contain information needed to use the GT-FPP/3 as a processing element, with both the ADA and C programming languages supported. Any changes deemed necessary from feedback on the original manual will also be included.

2.5.4. Materials Management System

Purpose

The purpose of the materials management system is to provide an organized, automated way to order PFP parts at the component, subsystem, and system levels. To do this, a database has been built using Borland International's Reflex database. The database is used to accumulate all needed parts for a particular PFP setup. All parts from each sub-assembly are summed into one ordering list for the purpose of ordering all similar parts together. This way fewer parts orders are generated reducing the possibility of overlooked or duplicated parts.

Database Form and Contents

The database is made up of sections called FIELDS. The FIELDS are shown in Table 2.1.

Table 2.1 Fields Used in Materials Management System

SUBASSEMBLY	Board or hardware piece name.
SUBASSE QTY	Quantity of sub-assembly per setup.
QTY PER ASSE	Quantity of part per sub-assembly.
PART NUM	Part number of particular part.
REFERENCE NUM	Reference number used in Technical Data Package.
VENDOR	Vendor who sells the particular part.
MANUFACTURER	Manufacturer of the particular part.
SS	Specifies if part is sole sourced or available from multiple vendors.
ITEM DESCRIPTION	Description of particular part.
ENGINEER	Engineer responsible for board or hardware piece.
UNIT PRICE	Price per unit piece.
TOTAL COST	Cost for multiple parts per 1 sub-assembly.
EXTENDED COST	Cost per multiple number of sub-assemblies.

Organization

The fields have been organized into two output formats, the PFP Materials List and the PFP Purchasing List. The formats are designed around specific output requirements.

The PFP Materials List format is used in the documentation process. It is formatted to best show item or

part information as referenced in the Technical Data Package. The Technical Data Package contains a parts list in this format for each assembly and sub-assembly in the PFP.

The Purchasing format is used in the purchasing process. It is formatted to sort all similar parts together, sort the vendors, and add the grand totals for a projected system cost .

Database Use

The Reflex Database system is a straightforward, easy to use, flat database. All the needed subassembly and parts breakdown are already intact and may be manipulated as shown by the following examples.

A. In order to get a materials list for documentation and board manufacturing purposes do the following:

    1. Choose the specified subassembly parts by using the filter command.

    2. Fill in the SUBASSE QTY (subassembly quantity) with the appropriate number.

    3. Print the contents generated by steps 1 and 2 in the PFP Materials List Format.

B. In order to accumulate all needed parts and prices for the purchase order process use the database contents generated in A1 and A2 and print contents in the Purchasing Format.

Database Modification

The Modification of parts, prices, and quantities in the database requires the following steps.

    1. Filter by PART NUMBER to find the part/s needing to be changed.

    2. Make modifications then remove filter.

    3. Repeat steps 1 and 2 until all modifications are made.

    4. Save the modified database.

The process required to add new parts to the database is as follows.

    1 Press the end key. This will bring cursor to a blank at the end of the database.

    2. Enter new data into all displayed fields.

    3. Repeat steps 1 and 2 until all new data is added.

    4. Run the Sort command.

    5. Save the modified database.

The process required to delete unwanted parts from the database is as follows.

1. Filter by PART NUMBER in the List View to find the part/s needing to be deleted.

2. Press F3 to select row containing the information to be deleted.

3. Press the delete key.

4. Repeat steps 1-3 until all specified parts are deleted.

5. Save the modified database.

2.6. PFP Training

A 3 day PFP training course was held in December 1989. The course was attended by 7 people. The course was divided into four separate sessions. Each session was followed by a short examination.

Session 1 contained a brief introduction to the parallel function processing approach. Basic parallel programming techniques were presented, including the methodology needed for partitioning a problem into its functional blocks. A small problem was presented and partitioned as an example.

Session 2 covered the hardware operation of the PFP. This included a fuctional description of the basic PFP architecture and how it works, its capabilities, and how to take the functional blocks and map them onto the machine. The basic issues of how to turn both the PFP and host on and off, how to start it, and where to access mass storage were also covered. The session also explained how to interpret the displays that are part of the machine in conjunction with program debug and system troubleshooting.

Session 3 covered programming the PFP. Topics included development and compilation of code for processing elements - including the use of special purpose I/O routines to interface with the crossbar and host computer, development of crossbar code - including syntax and compilation, how to integrate and load processing element codes with the crossbar code into a working program, and how to run the program.

Session 4 was held in the laboratory and consisted of dividing the attendees in groups of two and having them program two small problems on the PFP. The first problem was given in its complete form. The task was to copy it into the machine, compile it and run it. The second problem was given in block diagram form with the functional blocks outlined. Programmers developed their own code and ran it on the PFP.

The course size is limited by session 4. It requires that each group have enough access to the PFP to solve the programming problems in a reasonable amount of time. When using 1 PFP for training, the course size should be limited to around 10. No definite training schedule is planned. The course is available to be repeated when necessary.

2.7. PFP Testing

2.7.1. Reliability Testing and Temperature Analysis

A special technical report, "Parallel Function Processor Reliability Test" [9] has been written and delivered to USADC. The test consisted of running the PFP system diagnostics in an infinite loop, collecting thermocouple data from 32 different points on the system, and logging any system errors that

13

occurred in an output file. Temperature data was collected over a 103 hour period. Plots of 31 of these points are included in the report. One thermocouple did not work correctly, giving all negative temperatures.

The system was run with 2 full crossbars, 2 sequencers, 1 array interconnect link (i.e., one board on each crossbar) and 45 processors. All processors were the GT-FPP/3 floating point processor, which uses the most power and generates the most heat of all the processors currently supported. The remaining 17 processor slots were empty. The right processor bank was fully populated (31 processors, one array interconnect) with the remaining boards (14 processors, 1 array interconnect) on the left side.

Four diagnostic tests, T3, FPPMU, T2, and FUNCTION were run in a continuous loop for the whole week. Each diagnostic test is included in the Technical Data Package [6] Vol. 4. Results are given in two forms, per test and per processor.

2.7.2. GT-FPP/3 Accuracy Analysis

The GT-FPP/3 Floating Point Processor contains 10 hardware assisted functions. The hardware assisted functions are supported through the use of an add on board called the GT-FFS/1. The hardware assisted calculations are carried out much faster than the software algorithms used on most machines, thus adding to the GT-FPP/3's high performance. The functions currently supported by the GT-FFS/1 are:

1. Sine

2. Cosine

3. Tangent

4. Arcsine

5. Arccosine

6. Arctangent

7. Exponential

8. Natural logarithm

9. Reciprocal

10. Square root

Two programs were written and executed on the Parallel Function Processor (PFP) for each function. Both programs compared the values calculated by the GT-FPP/3 - GT-FFS/1 combination to the values calculated by the Intel 310 host computer. The first test generated the absolute difference between the numbers. The second test generated a relative error, using the number computed by the Intel 310 as the correct answer.

Two graphs were made for each function, one for absolute error and one for relative error. Although no major discrepancies were uncovered, full interpretation of the results is not yet complete. The graphs are included in Figures 2.1 through 2.20. After fully finishing all interpretations, the full analysis will be submitted to USADC as a special technical report.

2.8. Existing Systems

2.8.1. DETL PFPs

The Digital Emulation Technology Laboratory has three PFPs currently in use, two which are stand alone 32 processor units and one which is a 64 processor unit. A software development PFP, hosted by a Sun 386i computer, is primarily being used for developing a C compiler for the GT-FPP/3 floating point processor. The unit currently contains two iSBC386/12's and 35 GT-FPP/3s. An Ada to C translator is being bought so that both Ada and C languages will be available for the board. Several Intel iSBC386/20 processors are available to interchange with the GT-FPP/3 boards. Software development is under way to support the 386/20 within the Sun host environment.

A second Sun hosted 32 processor unit populated with GT-FPP/3 processors is being used for EXOSIM development work. This unit is serving as the initial test site for the Sun system software and C compiler.

A third unit, populated with 30 iSBC386/12 processors, is present in the DETL's secure laboratory. The unit is hosted by an Intel 310 and has been cleared for classified processing. The unit can support a maximum of 32 processors and is hosted by an Intel 310.

A PFP "test station" that can support up to 64 processing elements, originally put together from spare parts, is primarily used for testing new board assemblies, debugging defective assemblies, and for testing and integrating new PFP components. Currently, the unit is unpopulated but is available for testing and debugging when needed. The system is hosted by an Intel 310.

The thirty-two processor system that has been located in DETL's secure laboratory is currently being upgraded. Processor card cages are being retrofitted to provide more address capability. The system will be fully populated with iSBC386/12 processors and eventually hosted with a Sun 386i computer. When completed, the EXOSIM development work will be moved back to this machine.

2.8.2. KDEC PFP

A PFP prototype has been built specifically for use by the KDEC facility. The unit is capable of supporting 64 processors and 2 full crossbars. The unit will initially be shipped with 32 processors and 1 crossbar. The other crossbar and 32 processors may be added on site at a later date. The system is hosted by an Intel 310 computer. Programming languages available include FORTRAN, Pascal, C, and PL/M.

The processors currently installed in the system are the Intel iSBC286/12 boards which have been used in the CERL laboratory for the past 2 years. These processors, and accompanying software, are fully debugged and will provide a stable environment for the KDEC programmers to learn how to use the system. The other 32 processors, to be installed later, could be either 80386 or 80486 based (both code compatible with the 80286 based processors) or the GT-FPP/3 floating point processors.

The system was delivered on September 21, 1990. Support details are still being worked out.
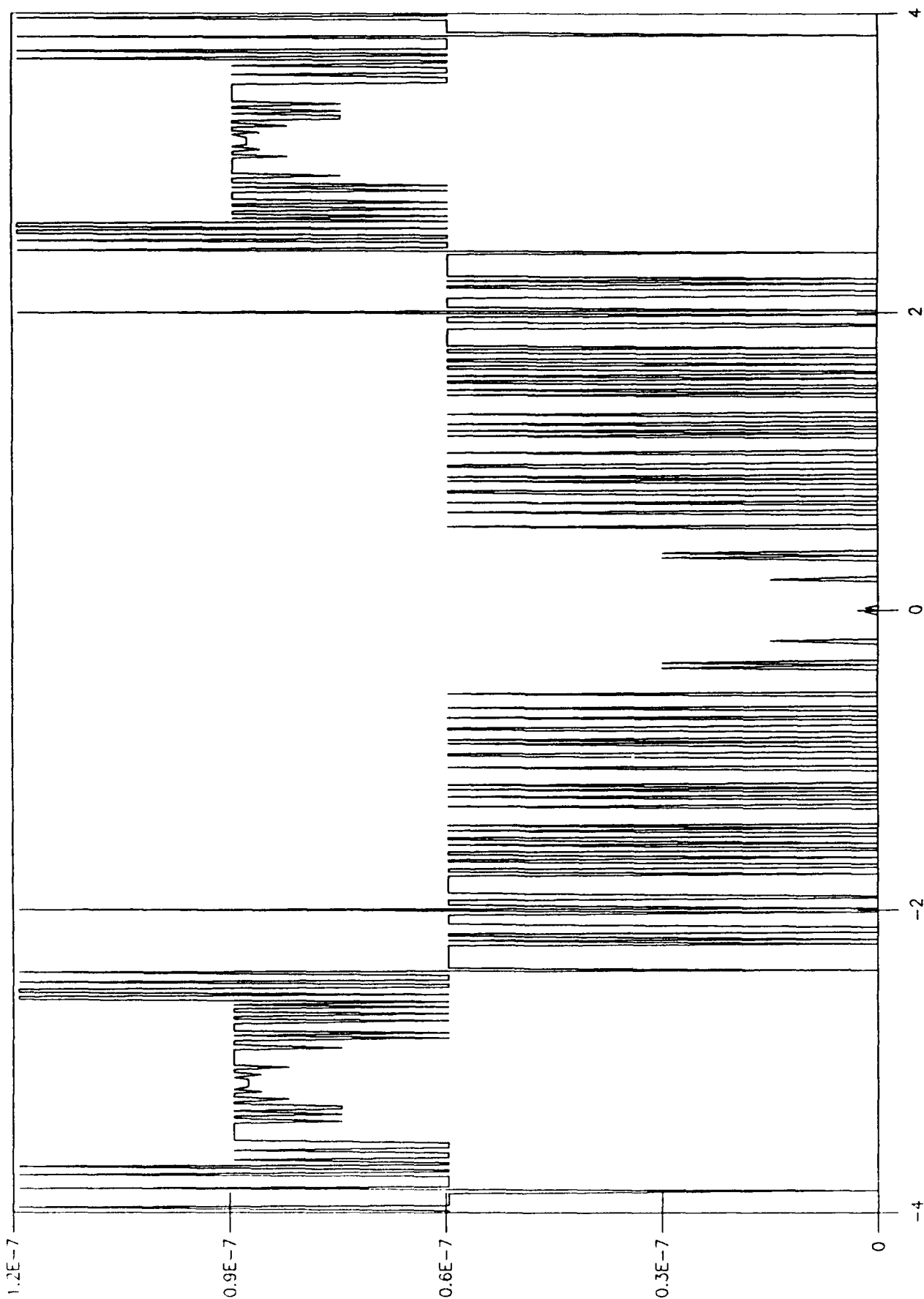
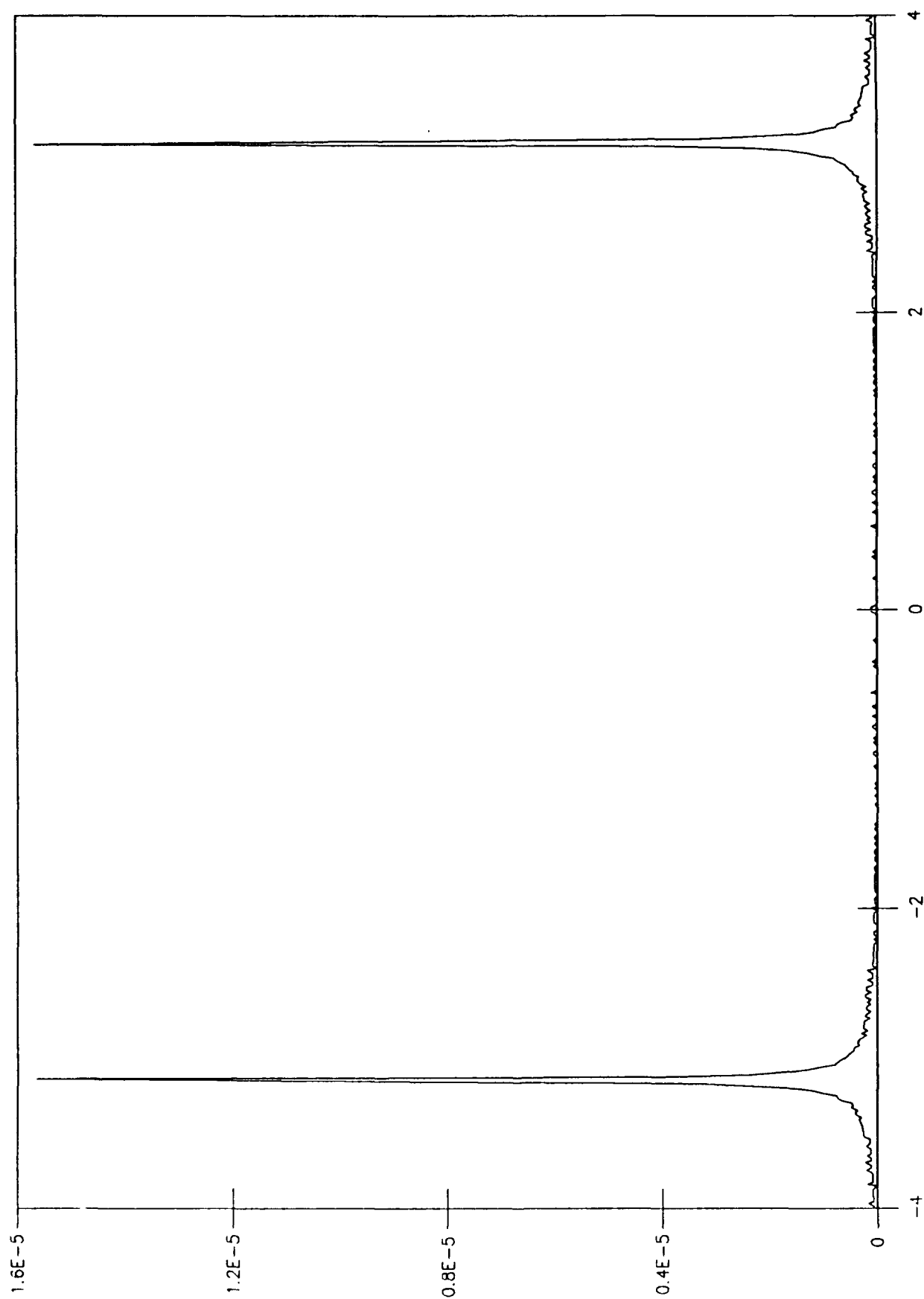Figure 2.1 Absolute error in sine function over reference angles.

16

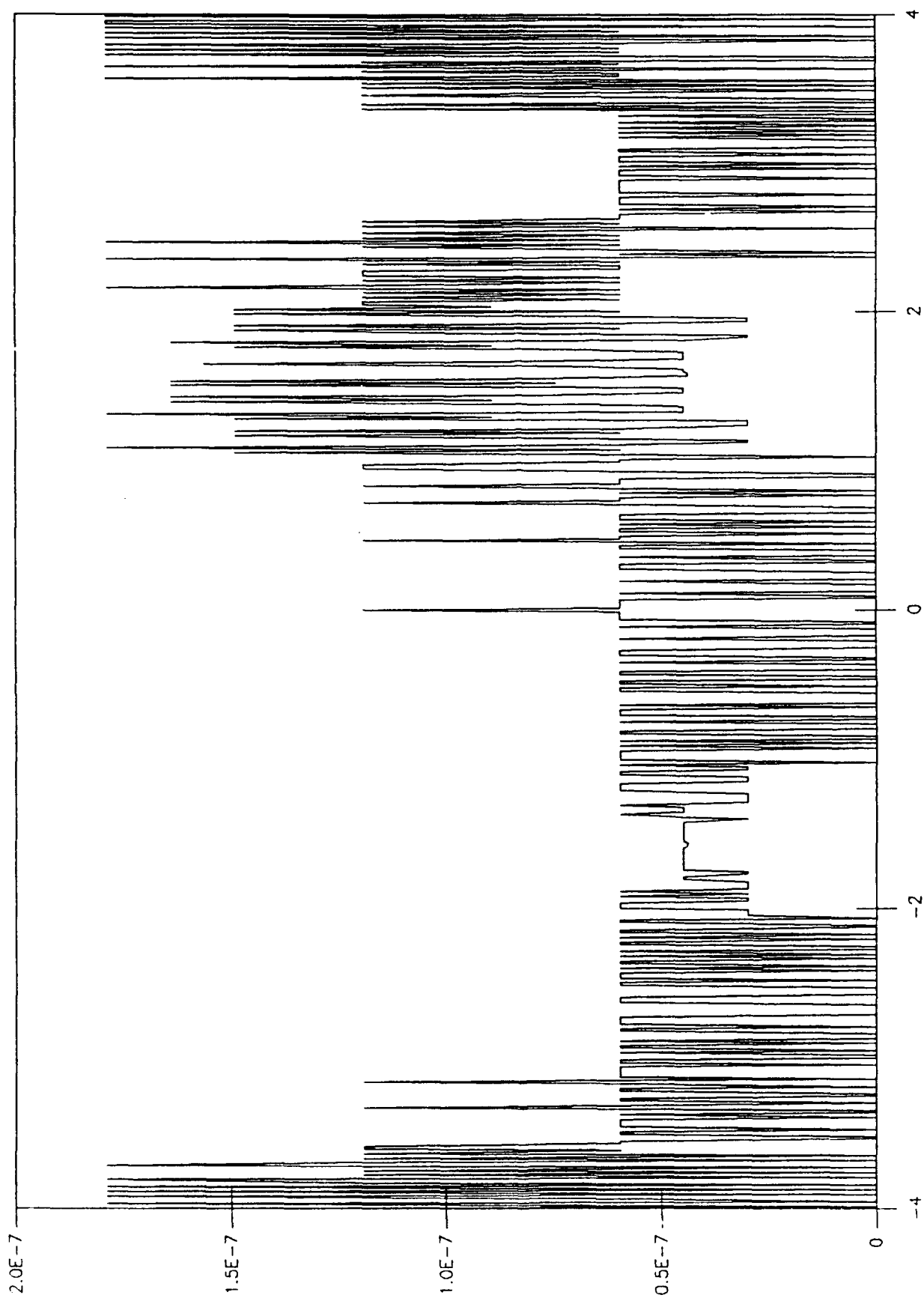Figure 2.2 Relative error in sine function over reference angles.

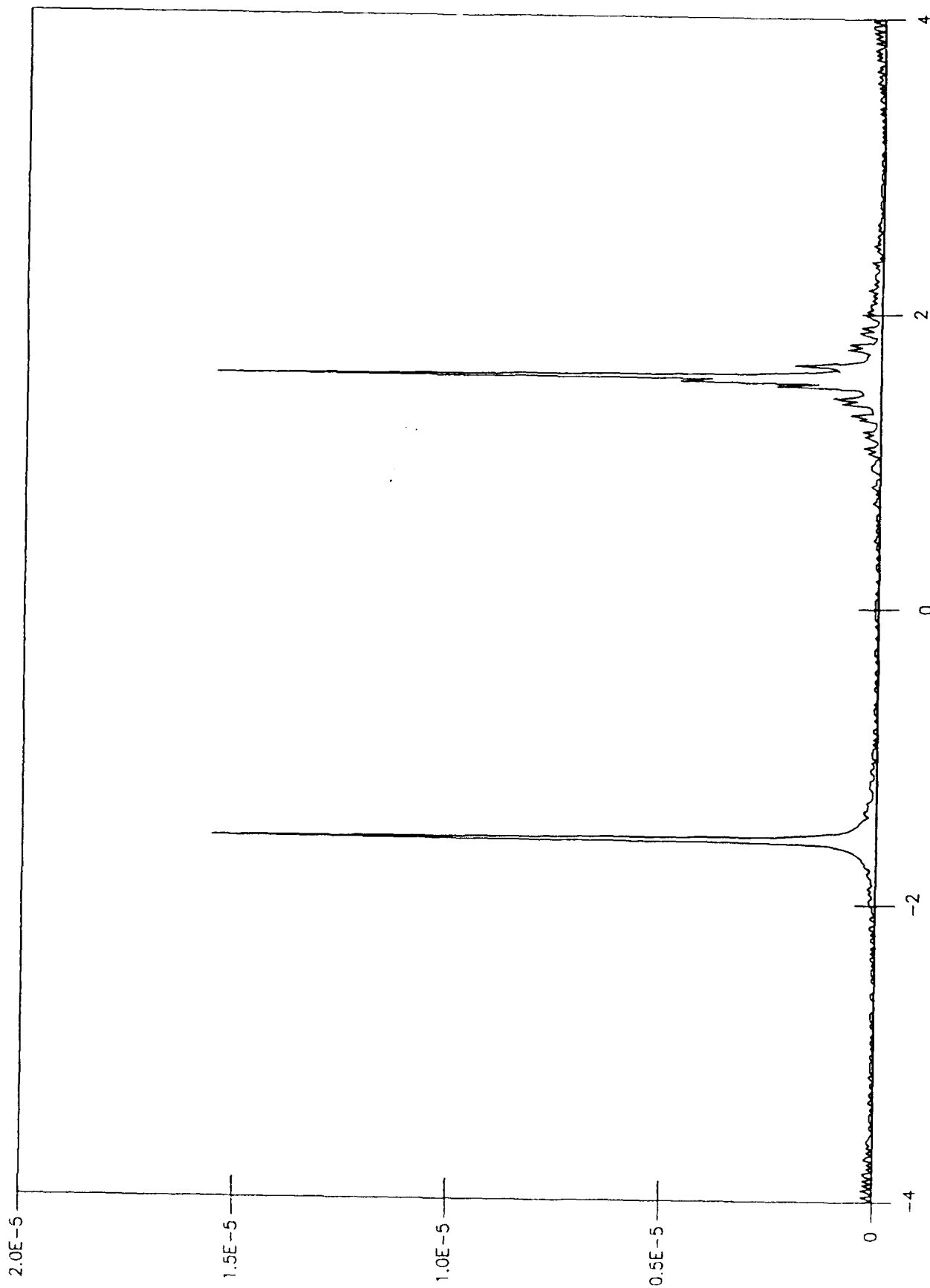Figure 2.3 Absolute error in cosine function over reference angles.

18

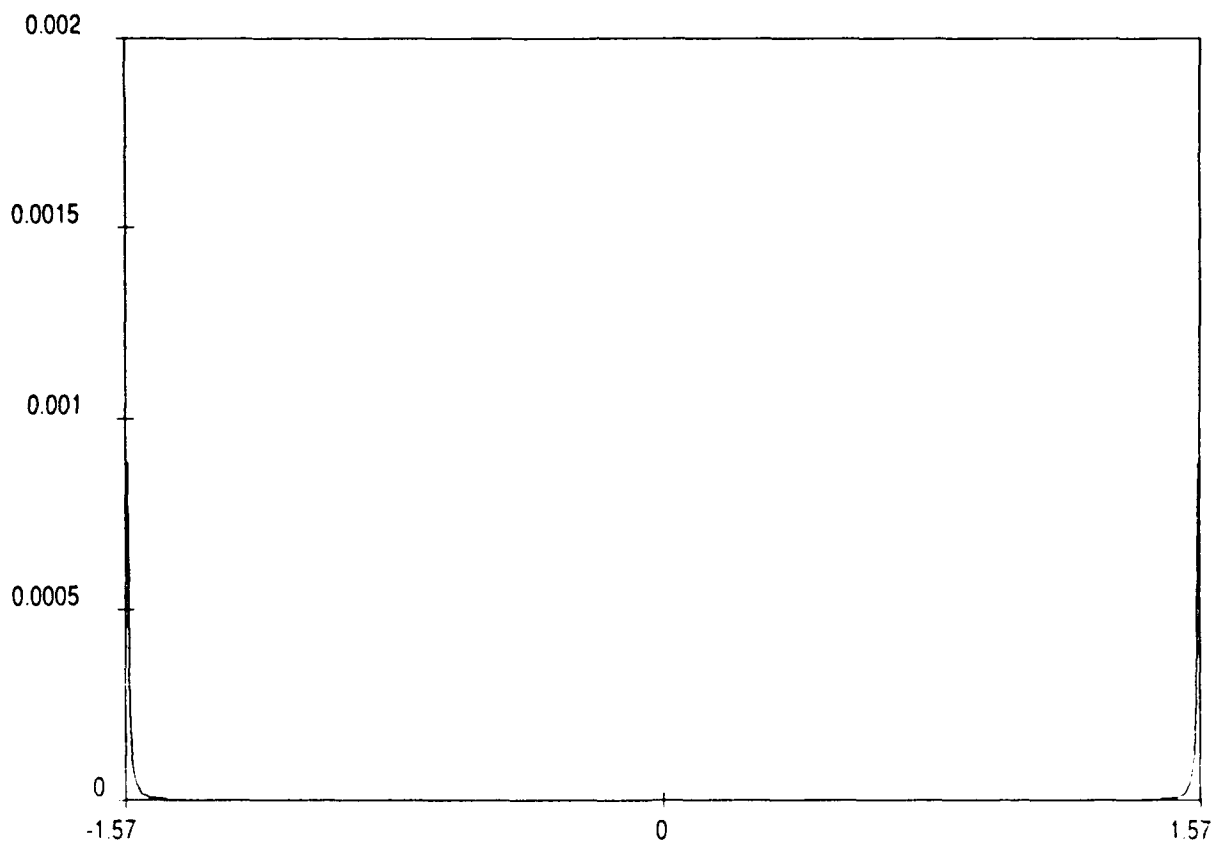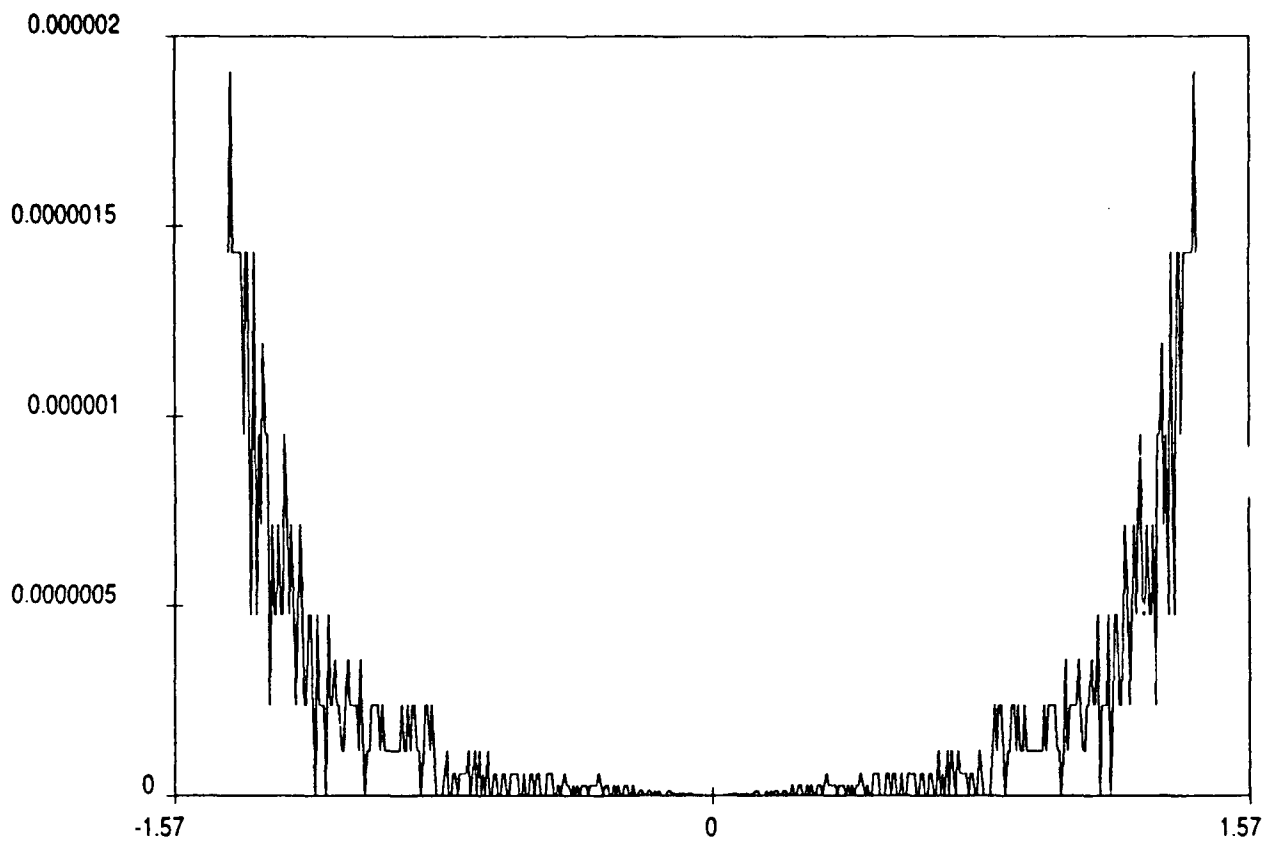Figure 2.4 Relative error in cosine function over reference angles.

Figure 2.5 Absolute error of tangent function at reference angles
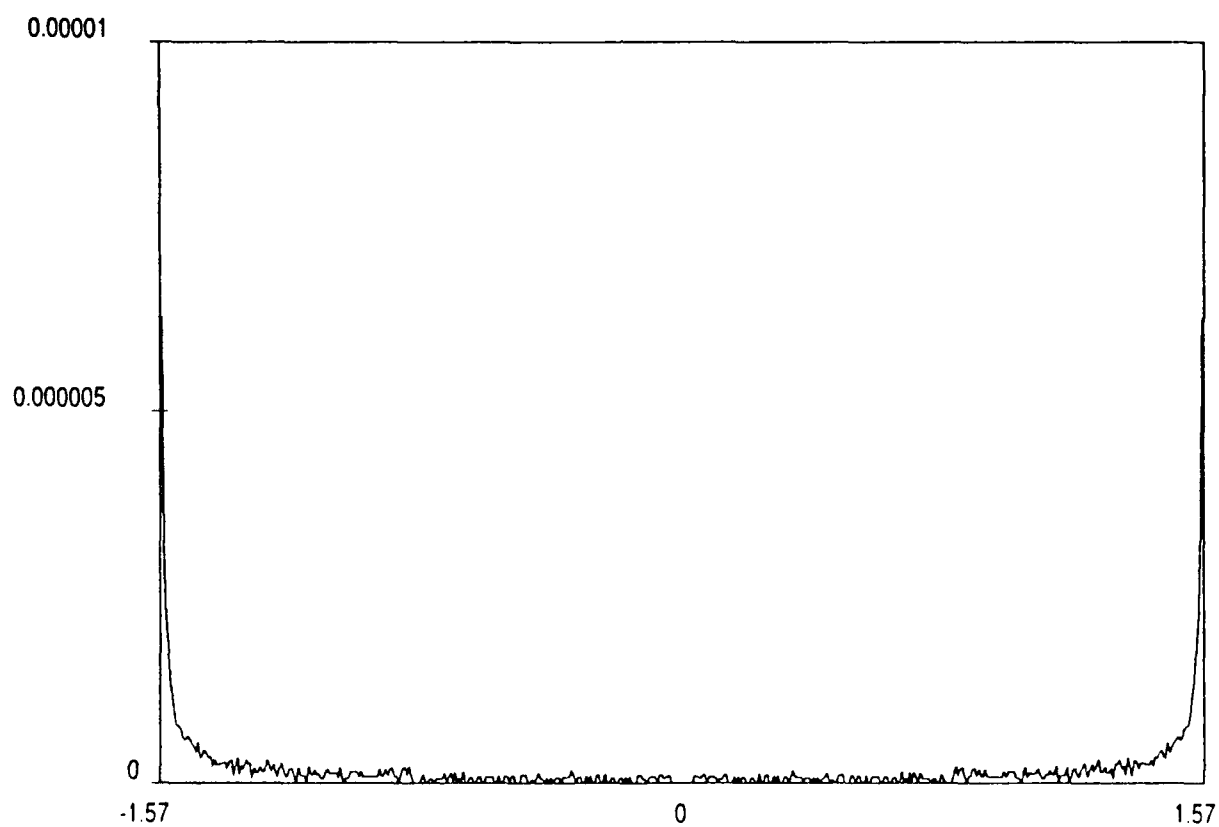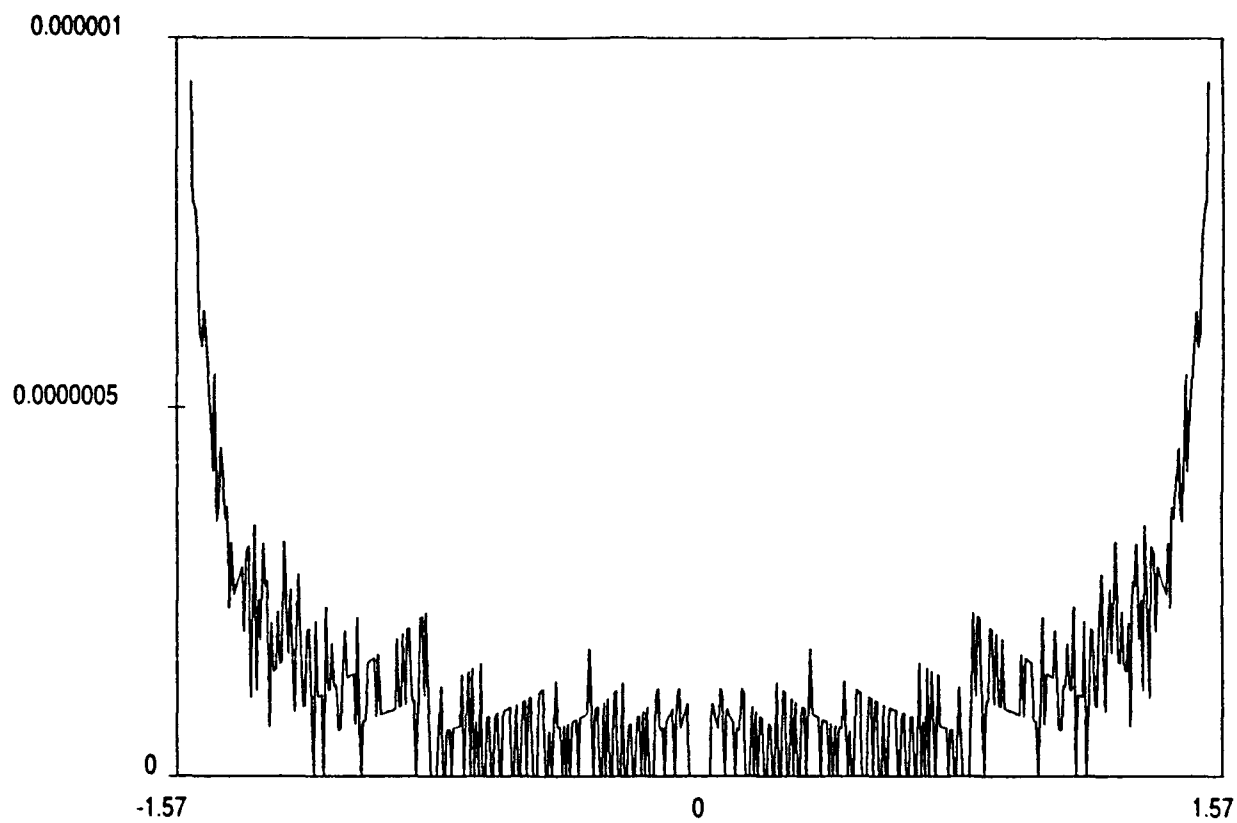(bottom) full view; (top) close up view of error.
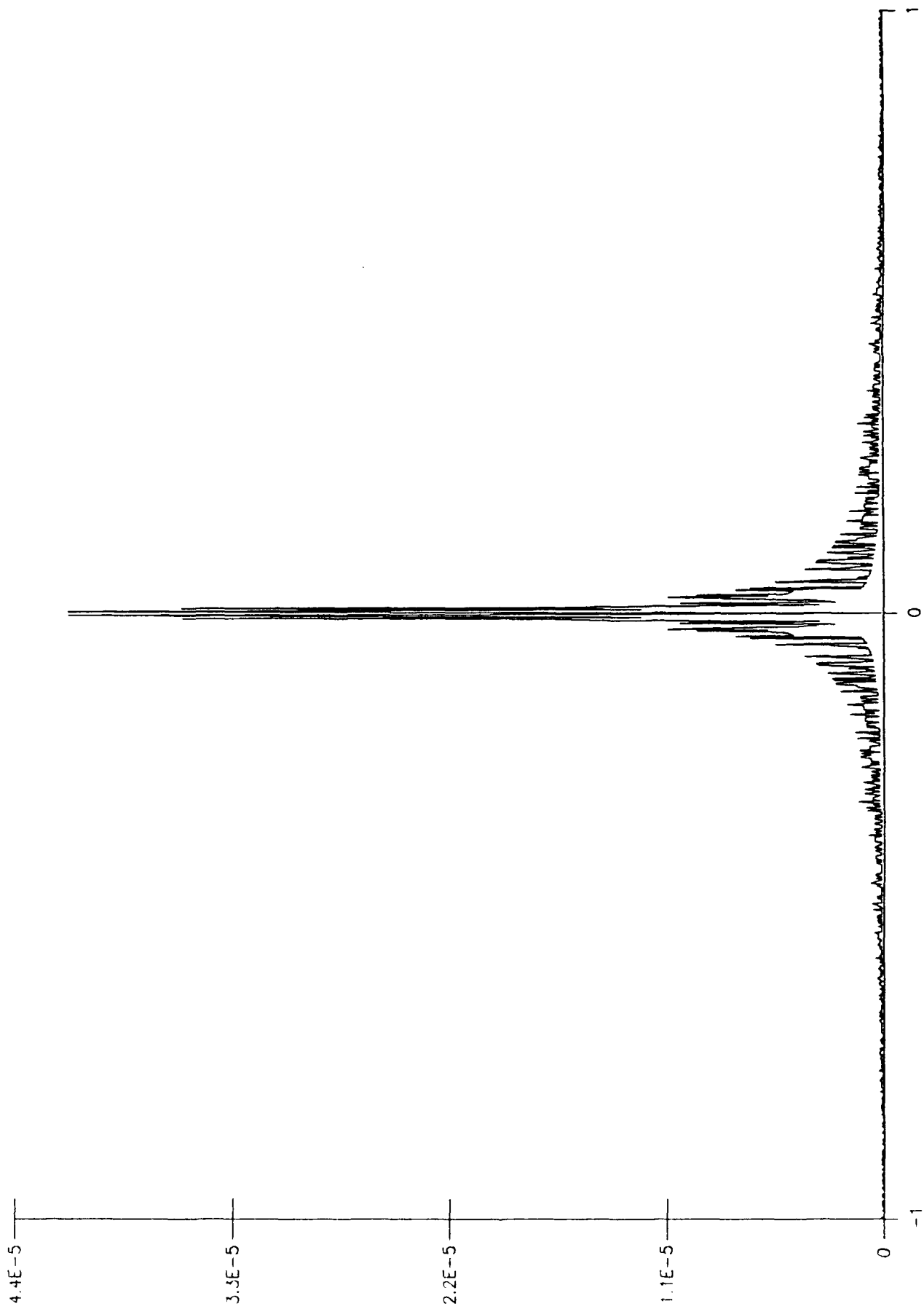
Figure 2.6 Relative error of tangent function at reference angles
(bottom) full view; (top) close up view of error.

21
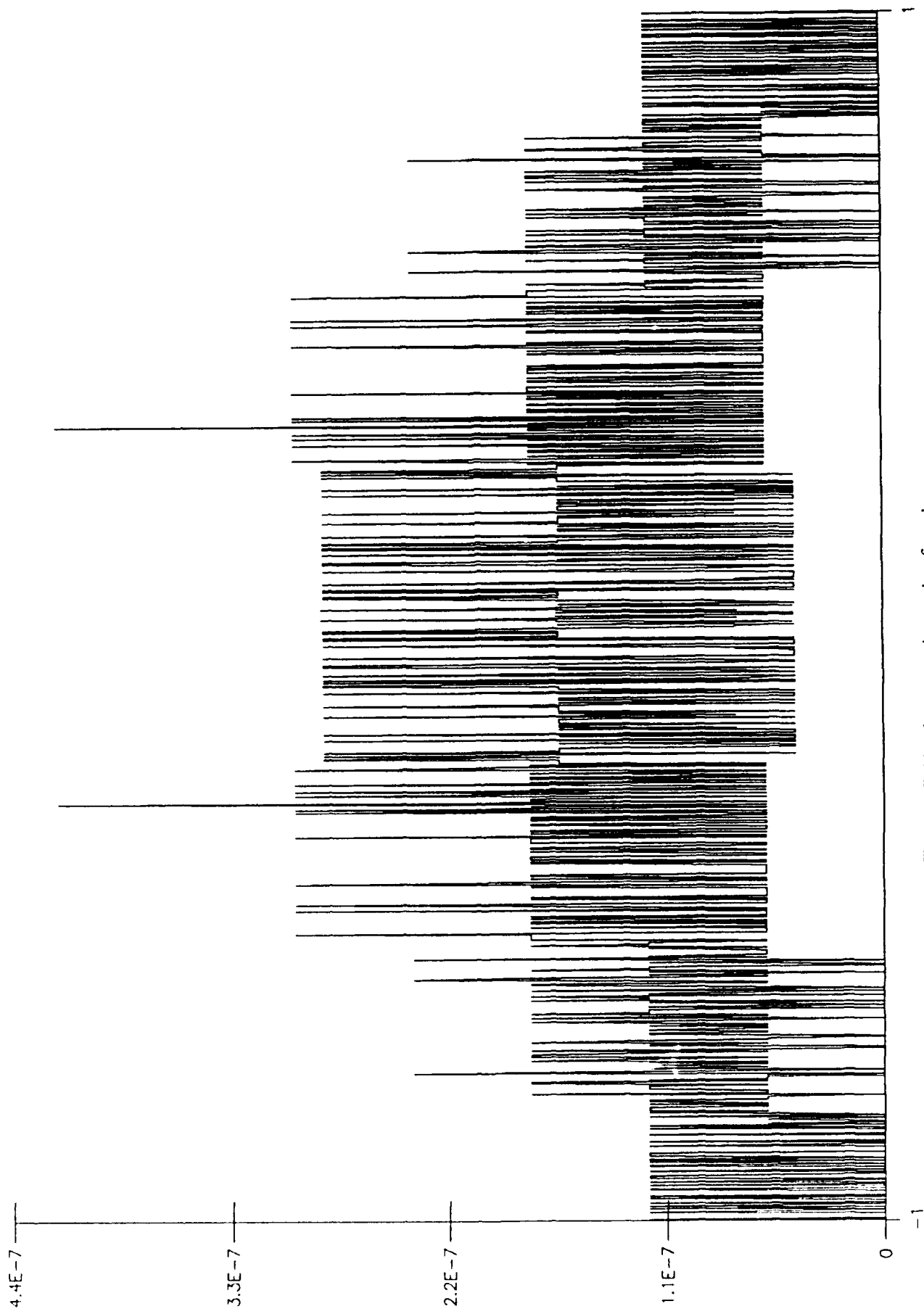
Figure 2.7 Absolute error in arcsine function.

Figure 2.8 Relative error in arcsine function.
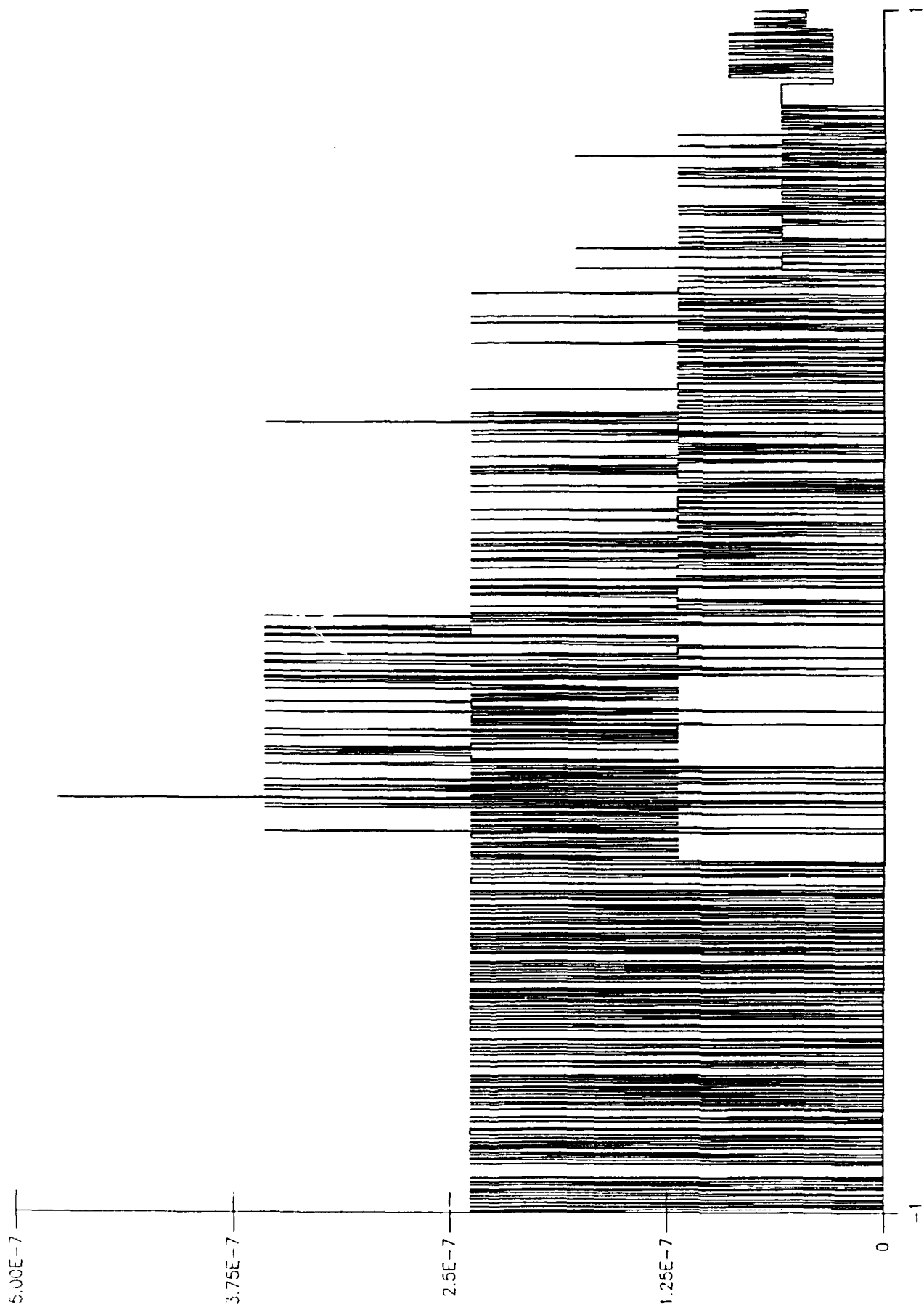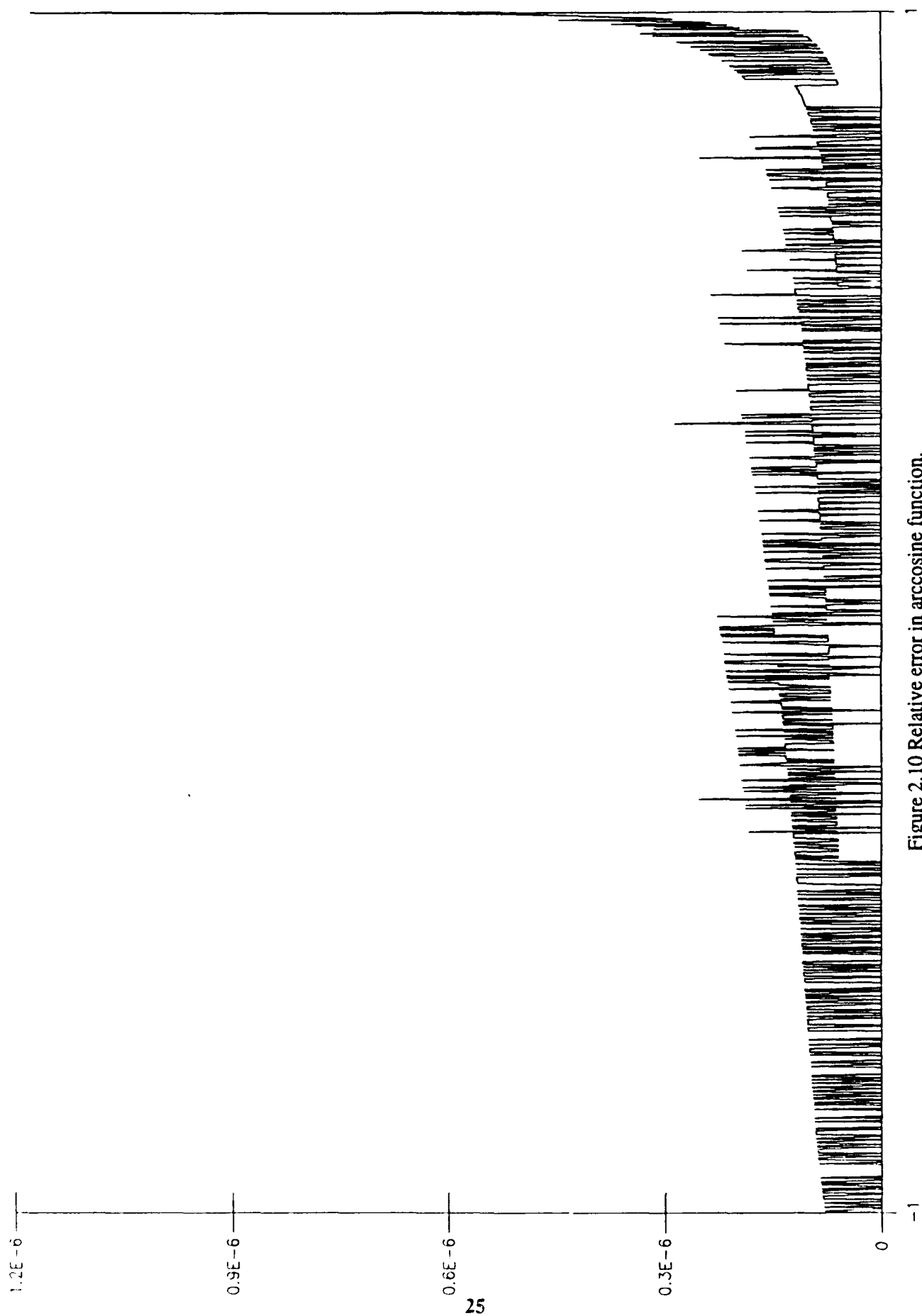
Figure 2.9 Absolute error in arccosine function.
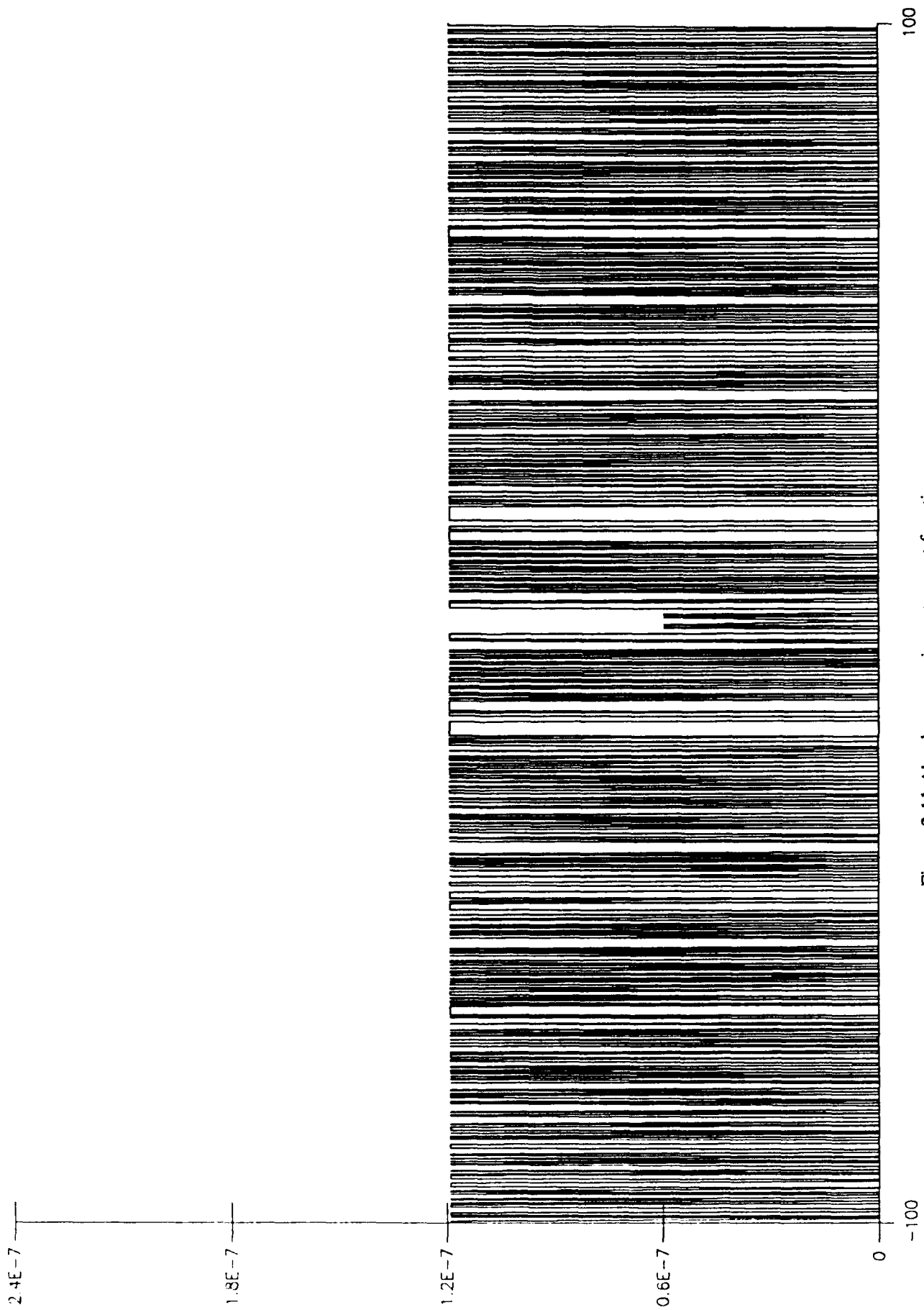
Figure 2.10 Relative error in arccosine function.

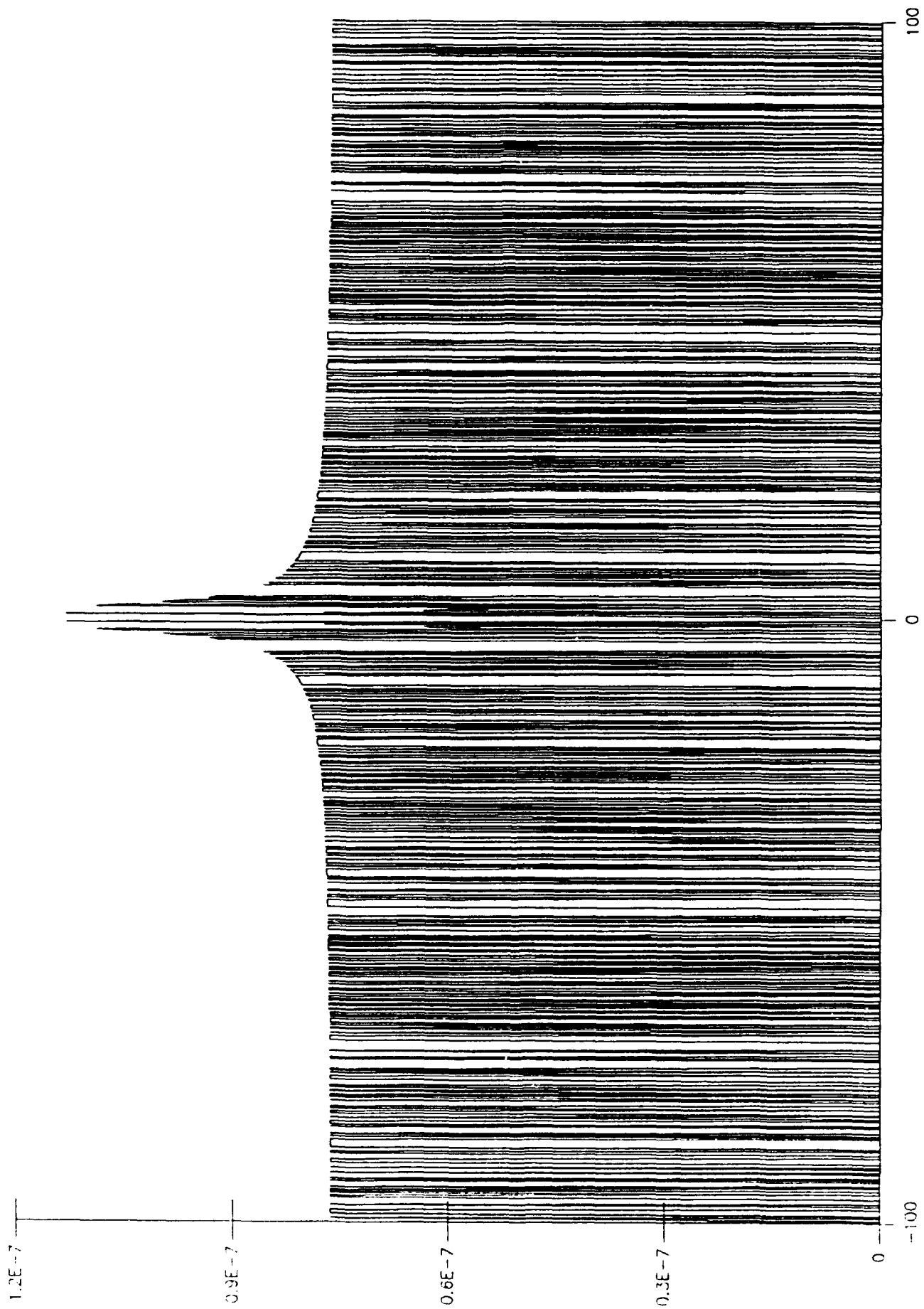Figure 2.11 Absolute error in arctangent function.

26

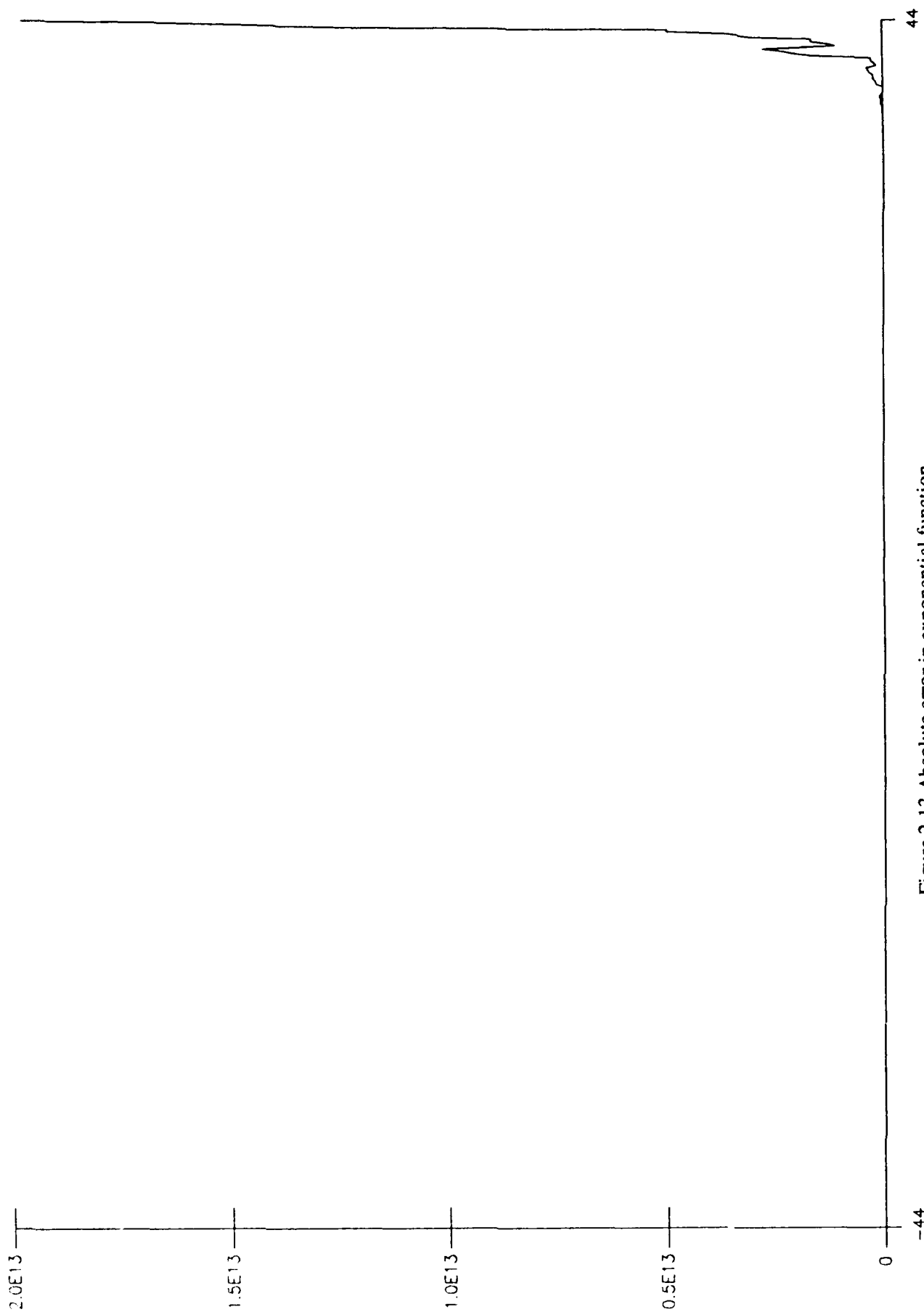Figure 2.12 Relative error in arctangent function.

27

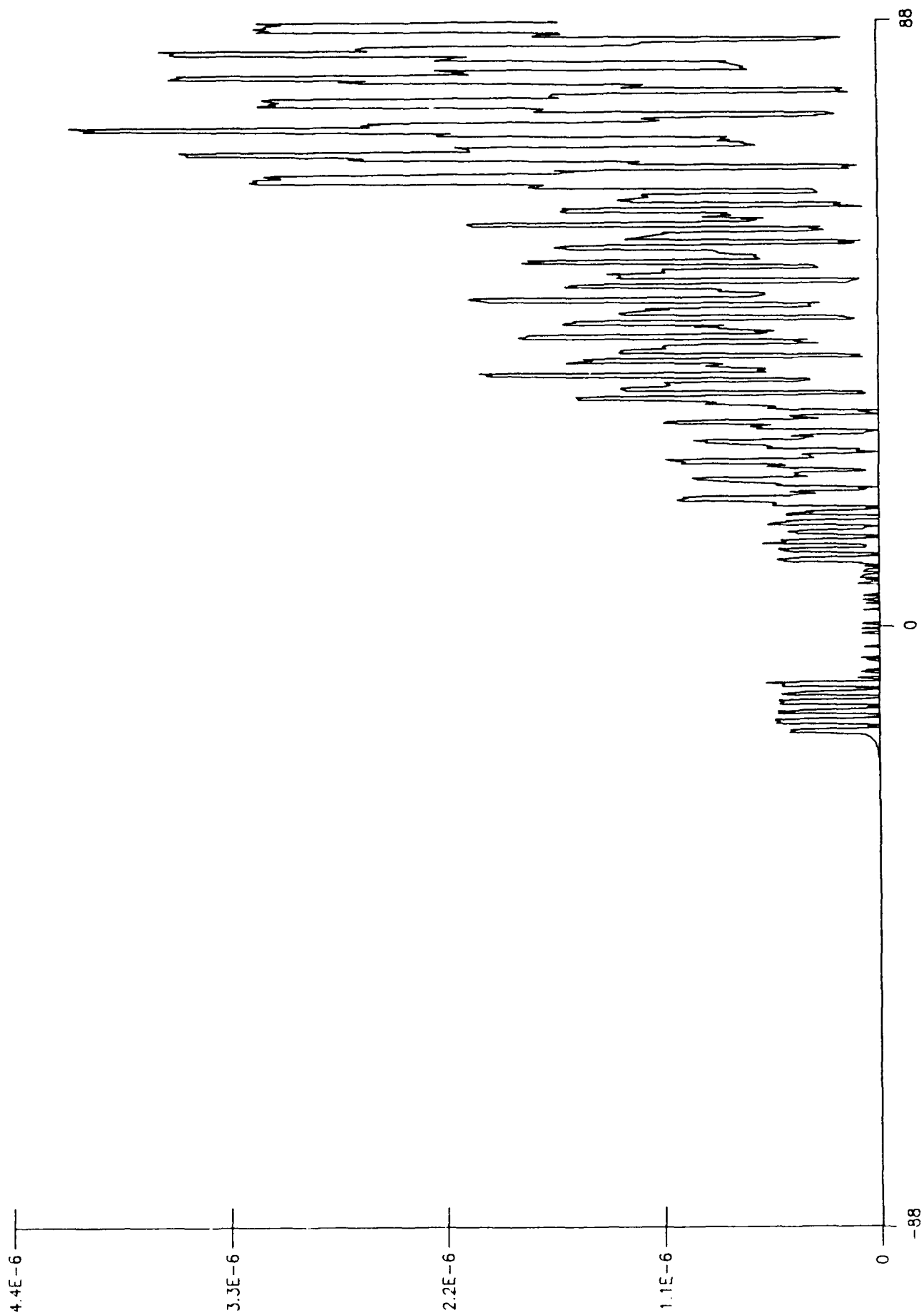Figure 2.13 Absolute error in exponential function.

Figure 2.14 Relative error in exponential function.

Figure 2.15 Absolute error in natural log function.

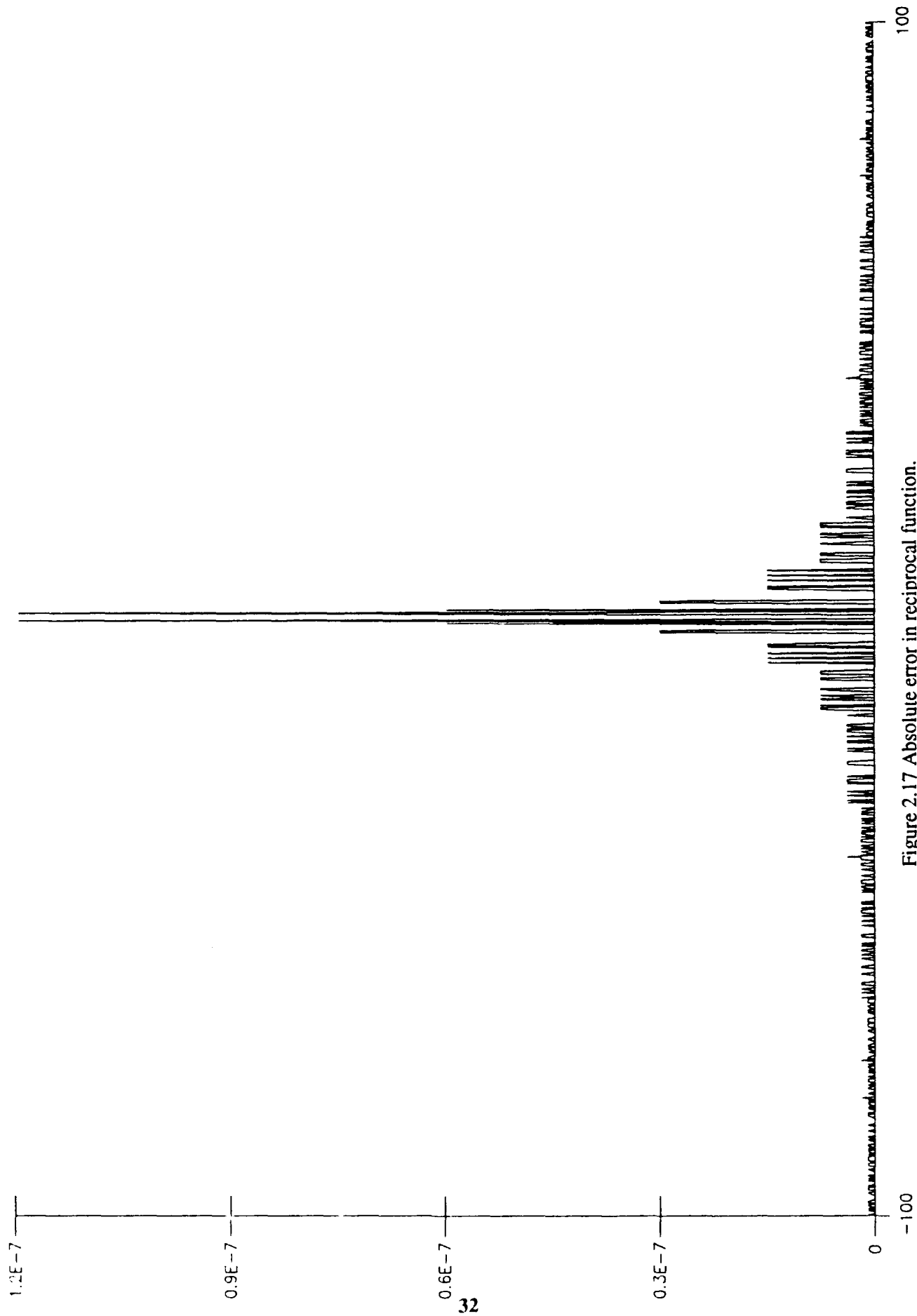Figure 2.16 Relative error in natural log function.

Figure 2.17 Absolute error in reciprocal function.
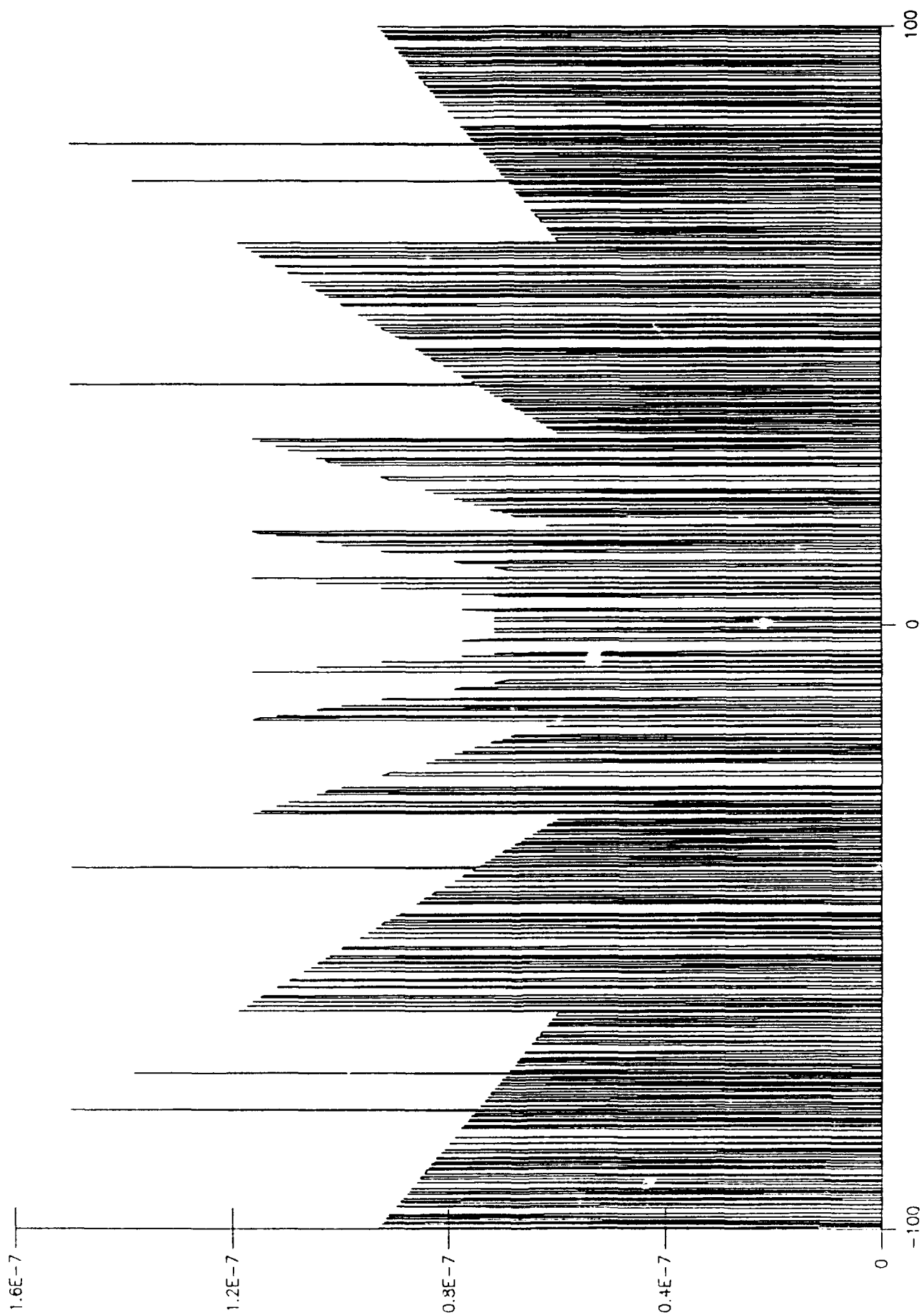
32

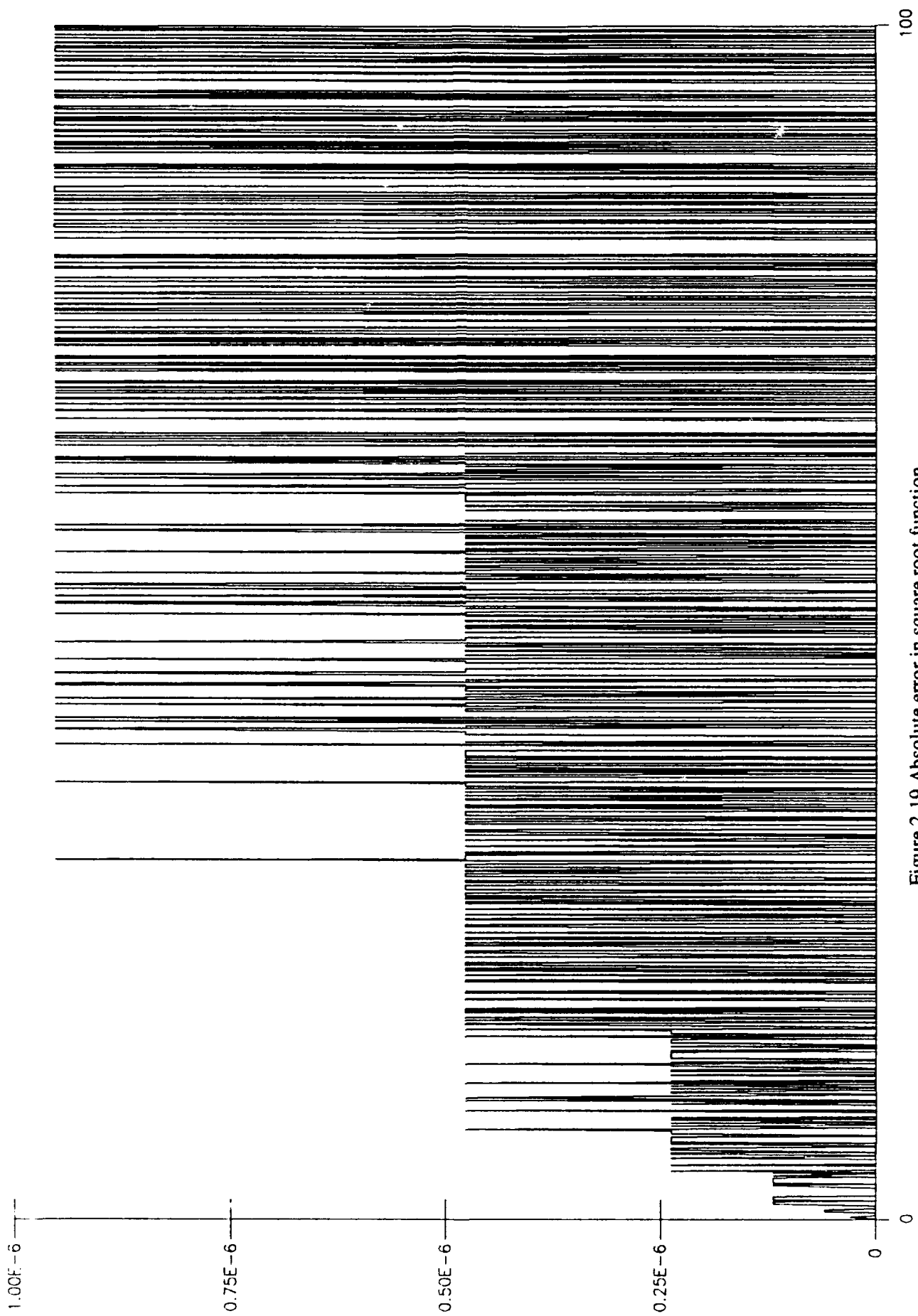Figure 2.18 Relative error in reciprocal function.

33

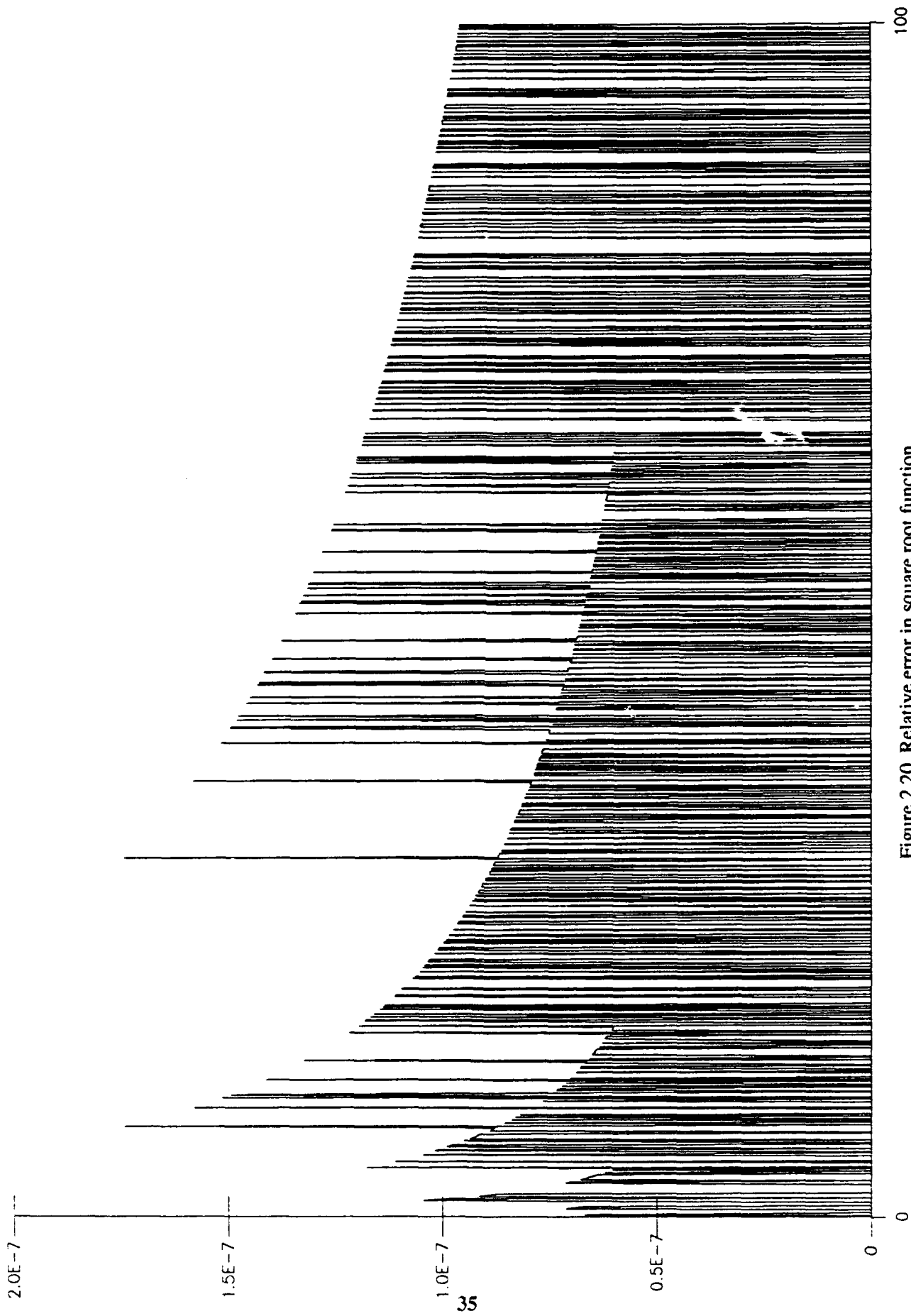Figure 2.19 Absolute error in square root function.

34

Figure 2.20 Relative error in square root function.

35

2.0E – 7

1.5E – 7

1.0E – 7

0.5E – 7

0

0

100

3. Performance Benchmarks

Performance benchmarks have been performed periodically over the span of this contract. They are mostly simulation programs representative of the type of applications the PFP has been designed for. The times from some of the earlier ones were taken on earlier versions of the system with slower processors, so the times may no longer be competitive. Time and manpower limitations have limited us from running every benchmark on our latest hardware, however, each benchmark was partitioned and run successfully in a parallel form. The most recent benchmark is the Spinning Missile Simulation. The EXOSIM simulation is the most complex simulation attempted yet, and currently being worked on. It should be completed this year.

3.1 Spinning Missile Simulation

The 6-DOF Spinning Missile Simulation is a program used by the EAI Corporation to benchmark their SIMSTAR computer system with the objective to run the program in real time. The source code was obtained from EAI, along with the correct results, and programmed on the PFP. The simulation was run on a system populated with iSBC286/12 processors and another populated with GT-FPP/3 processors. The GT-FPP/3 based system ran the benchmark several times faster than real time. Details are available in the FY89 final report [1] Vol 1, p. 56 and Vol. 4, Appendix E.

3.2 Target Initialization Benchmark

The Target Initialization benchmark was taken from the target initialization code from the SKEW (SAIC, Eglin Air Force Base) and extracted for the PFP. The objective was to show how the code moved from a serial machine to a parallel machine. Using the iSBC286/12 processors, the code was run twice as fast as real time. Answers matched the results from a VAX 11/780 running the serial code. Details are available in the FY89 final report [1] Vol 1, p. 55 and Vol. 4, Appendix D.

3.3 Satellite Attitude Control System

The Sattellite Attitude Control System was given in an application report published by Applied Dynamics International. It was used as a benchmark for the Applied Dynamics AD100 simulation computer. The same system was programmed on the PFP using the same methodology. Details are available in the FY87 final report [5] Vol. 2, p. 21 and Vol. 3, Appendix J. The interesting point to note is that the entire AD100 system has about 1/3 the throughput of a single GT-FPP/3 processor.

3.4 High-Endoatmospheric Simulation

Two versions of a 3-DOF High Endoatmosheric simulation were developed for the PFP. The first version was a linear model and the second was a non-linear model. Details on the models and the results from the PFP are included in the FY87 final report [5] Vol. 2, pp. 20 - 21.

3.5 Exoatmospheric Simulation

A linear 3-DOF exoatmosheric model was developed as a first prototype for ERIS. The model was programmed on the PFP using 30 processors. Details and results are included in the FY87 final report [5] Vol. 2, p21 and Vol. 3, Appendix I.

4. References

1. Collins, Thomas R., et al, "Final Technical Report", Contract No. DASG60-85-C-0041, Volume 1, Georgia Institute of Technology, July 20, 1989.

2. McQuade M. R.,"A Macrologic Array for Simulation", PhD dissertation, Georgia Institute of Technology, Atlanta, Ga., March 1980.

3. Hamblen, J. O., "The Design and Performance of a Parallel Computer Architecture for Simulation", PhD dissertation, Georgia Institute of Technology, Atlanta, Ga., February 1984.

4. Pitts, Richard M., "Parallel Function Processor Programmer's Manual", Special Technical Report STR-0142-90-006, Contract No. DASG60-89-C-0142, Georgia Institute of Technology, January 23, 1990.

5. Alford, C. O., et al, "Final Technical Report", Contract No. DASG60-85-0041, Volumes 1-3, Georgia Institute of Technology, November 10, 1987.

6. "Parallel Function Processor Technical Data Package", Contract No. DASG60-89-C-0142, Volumes 1-4, Georgia Institute of Technology, March 31, 1990.

7. Alford, C. O., et al, "Hardware-in-the-Loop Testing of KEW Flight Processors", Proceedings of the 1990 Summer Computer Simulation Conference, Society for Computer Simulation, July 18, 1990, pp 938-943.

8. Woods, Michael B., "Parallel Function Processor Hardware Operation", Special Technical Report STR-0142-90-005, Contract No. DASG60-89-C-0142, Georgia Institute of Technology, January 23, 1990.

9. Woods, Michael B, Richard M. Pitts, and James N. Hardage, "Parallel Function Processor Reliability Test", Special Technical Report STR-0142-90-004, Contract No. DASG60-89-C-0142, Georgia Institute of Technology, January 23, 1990.